Robust computational intelligence techniques for visual information processing

UNIVERSIDAD DE MÁLAGA

TESIS DOCTORAL

Karl-Khader Thurnhofer Hemsi

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Enero 2021

AUTOR: Karl Thurnhofer Hemsi

https://orcid.org/0000-0001-6519-1213

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

Robust computational intelligence techniques for visual information processing

Memoria que presenta para optar al título de Doctor por la Universidad de Málaga

Karl-Khader Thurnhofer Hemsi

Dirigida por los Doctores

Ezequiel López Rubio
Núria Roé Vellvé

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Enero 2021

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña KARL-KHADER THURNHOFER HEMSI
Estudiante del programa de doctorado TECNOLOGÍAS INFORMÁTICAS de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: ROBUST COMPUTATIONAL INTELLIGENCE TECHNIQUES FOR VISUAL INFORMATION PROCESSING

Realizada bajo la tutorización de EZEQUIEL LÓPEZ RUBIO y dirección de EZEQUIEL LÓPEZ RUBIO Y NÚRIA ROÉ VELLVÉ (si tuviera varios directores deberá hacer constar el nombre de todos)

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 25 de ENERO de 2021

Fdo.: KARL-KHADER THURNHOFER HEMSI

Edificio Pabellón de Gobierno. Campus El Ejido. 29071
Tel.: 952 13 10 28 / 952 13 14 61 / 952 13 71 10
E-mail: doctorado@uma.es

UNIVERSIDAD DE MÁLAGA

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

El Dr. D. Ezequiel López Rubio, Catedrático de Universidad perteneciente al área de Ciencia de la Computación e Inteligencia Artificial de la E.T.S. de Ingeniería Informática de la Universidad de Málaga, y la Dra. D.ª Núria Roé Vellvé, perteneciente al Centro de Investigación Biomédica en Red en Bioingeniería, Biomateriales y Nanomedicina (CIBER-BBN),

Certifican que,

D. Karl-Khader Thurnhofer Hemsi, Graduado en Ingeniería del Software y Licenciado en Matemáticas, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo su dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

Robust computational intelligence techniques for visual information processing

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo. Y para que conste a efectos de lo establecido en la legislación vigente, autorizamos la presentación de este trabajo en la Universidad de Málaga.

Málaga, Enero de 2021

Fdo.: Dr. Ezequiel López Rubio

Fdo.: Dra. Núria Roé Vellvé

A mi madre,
por toda su paciencia conmigo,
a tí que entregaste todo por mí,
y te quiero como a nadie.

Acknowledgments

I have been studying all my life. Since I started when I was three years old, at school, high school, and university, I have been acquiring knowledge that, over time, has proven useful both for my professional career and for personal life. And during all this time, my mother has raised, taken care of, and helped me in everything I needed. Therefore, my first dedication is to her, who full of patience, always hopes that I will be happy with my decisions.

This Ph.D. thesis has been an academic and research challenge. On the one hand, an unknown path began for me, which I saw from afar as something that was not for me, too complex. That is why the first thanks are focused on my thesis directors, Ezequiel and Nuria, with whom none of this would have been possible. In chronological order, Nuria gave me the necessary push to start this battle and taught me to be a good researcher, and Ezequiel took me into his arms without knowing me even visually, to transmit all his knowledge to me. I want to highlight my gratitude to Nuria for her effort to help me start this project when the professional situation was worrying. Also, to unconditionally support me in all my decisions. Regarding Ezequiel, I never thought this was going to go so far the first day we met, and that he was such a good person, especially in the human aspect, helping me in the worst times. And most importantly, I think that thanks to him, I have discovered a profession that I love, and that motivates me to improve it.

However, this thesis was also the beginning of the greatest joys of my life: the first trips, the great coworkers with whom we have set up a successful soap opera, the stays at other universities that made me grow as a person and as a researcher, the wonderful people I have met...

Starting from my professional side, I want to thank Enrique and Rafa for their altruistic collaboration with me to achieve the objectives of my predoctoral training. I always keep his work and teachings in mind. And on the other hand, thank my colleagues Miguel, Jesús, Jorge, Julio, and Rosa, for being part of what I consider my family. Especially to Miguel for being the first one I met, and who has been a mentor and also a Moisés, opening the bureaucratic path. To the others, I hope you are clear about who The Doctor is. I have not had the pleasure of meeting Pepe, but I am sure that his legacy remains in all those mentioned here. I would not want to forget

either the stays I have made in Ann Arbor and Leicester, as well as the people I have met there. Starting with Kayvan and his team, who helped me integrate and improve as a researcher, and ending with David and Lipika, who welcomed me as a relative. I want to thank you for your interest in any aspect that influenced my doctorate. Likewise, I appreciate the University of Michigan and De Montfort University for hosting me and providing me with the necessary resources to carry out my work. I don't want to end this paragraph without mentioning Aitor, a great friend with whom I toured part of the USA.

From the personal side, I would like to start by thanking Fran and José, my career colleagues, for their friendship all those years. I continue with the UNCA team: Marcelo, Lupe, Diana, María José, Javier, Juan Miguel, and Lisa. With them, I started scientific research, but I keep its human side. I have never met more fantastic people in my life. They supported me in the worst moments of my life from the leftover. In this regard, I want to thank Diana immensely for her particular presence in this universe. I couldn't have done this without you. I also do not forget those people who were with my most intimate side. Carmen, you arrived just at the most necessary moment. Thank you. And thanks for being the way you are, don't change, everything will come. Just lower that level of madness a little bit. Teresa, I loved you very much, and I spent the best year of my life with you, and even if you want to go a different way, I will never forget you. I hope you find what fulfills you and makes you 100% happy.

Now finally, I finish this long stage and start another one full of enthusiasm and desire to work. Thank you all.

Financial Support

This Ph.D. thesis is partially supported by the Ministry of Economy and Competitiveness of Spain under grants TIN2014-53465-R, TIN2016-75097-P and PPIT.UMA.B1.2017. It is also partially supported by the Ministry of Science, Innovation and Universities of Spain under grant RTI2018-094645-B-I00, and by the Autonomous Government of Andalusia (Spain) under grants TIC-6213 and UMA18-FEDERJA-084. All of them include funds from the European Regional Development Fund (ERDF). I thankfully acknowledge the computer resources, technical expertise, and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga, and also the grants of the Universidad de Málaga itself. It has also been supported by the Biomedical Research Institute of Málaga (IBIMA). Finally, the grateful acknowledgment of the support of NVIDIA Corporation with the donation of two Titan X GPUs used for this research. Karl Thurnhofer-Hemsi (FPU15/06512) was supported by a Ph.D. scholarship from the Spanish Ministry of Education, Culture and Sport under the FPU program.

Abstract

This Ph.D. thesis is about image processing by computational intelligence techniques. Firstly, a general overview of this book is carried out, where the motivation, the hypothesis, the objectives and the methodology employed are described. The use and analysis of different mathematical norms will be our goal. After that, state of the art focused on the applications of the image processing proposals is presented. In addition, the fundamentals of the image modalities, with particular attention to magnetic resonance, and the learning techniques used in this research, mainly based on neural networks, are summarized. To end up, the mathematical framework on which this work is based on, ℓ_p-norms, is defined.

Three different parts associated with image processing techniques follow. The first non-introductory part of this book collects the developments which are about image segmentation. Two of them are applications for video surveillance tasks and try to model the background of a scenario using a specific camera. The other work is centered on the medical field, where the goal of segmenting diabetic wounds of a very heterogeneous dataset is addressed.

The second part is focused on the optimization and implementation of new models for curve and surface fitting in two and three dimensions, respectively. The first work presents a parabola fitting algorithm based on the measurement of the distances of the interior and exterior points to the focus and the directrix. The second work changes to an ellipse shape, and it ensembles the information of multiple fitting methods. Last, the ellipsoid problem is addressed in a similar way to the parabola.

The third part is exclusively dedicated to the super-resolution of Magnetic Resonance Images. In one of these works, an algorithm based on the random shifting technique is developed. Besides, we studied noise removal and resolution enhancement simultaneously. To end, the cost function of deep networks has been modified by different combinations of norms in order to improve their training.

Finally, the general conclusions of the research are presented and discussed, as well as the possible future research lines that are able to make use of the results obtained in this Ph.D. thesis.

Contents

List of Abbreviations

BC Bhattacharyya Coefficient.

CIMES Medical Research Center of the University of Málaga.

CNN Convolutional Neural Network.

CPU Central Processing Unit.

CT Computerized Tomography.

DCSRN Densely Connected Super-Resolution Network.

DICOM Digital Imaging and Communication in Medicine.

DL Deep Learning.

GD Gradient Descent.

GPU Graphic Processing Unit.

GT Ground Truth.

HCP Human Connectome Project.

HR High-Resolution.

IBSR Internet Brain Segmentation Repository.

IoU Intersection on Union.

LR Low-Resolution.

MCC Matthews Correlation Coefficient.

MFT Median Filter Transform.

MFT3D 3D Median Filter Transform.

MISR Multi-Image Super-Resolution.

MR Magnetic Resonance.

MRI Magnetic Resonance Imaging.

MSE Mean Squared Error.

NIfTI Neuroimaging Informatics Technology Initiative.

NLM Non-Local Means.

NYU New York University.

OASIS Open Access Series of Imaging Studies.

PET Positron Emission Tomography.

PSNR Peak Signal-to-Noise Ratio.

PTZ Pan-Tilt-Zoom.

RGB Red-Green-Blue.

RMSO Root Mean Squared Orthogonal.

ROC Receiver Operating Characteristic.

SGD Stochastic Gradient Descent.

SISR Single-Image Super-Resolution.

SNR Signal-to-Noise Ratio.

SPECT Single Photon Emission Computed Tomography.

SR Super-Resolution.

SRCNN Super-Resolution Convolutional Neural Network.

SRCNN3D 3D Super-Resolution Convolutional Neural Network.

SSIM Structural Similarity Index Measure.

SVM Support Vector Machine.

WCS Weighted Chebyshev Scalarization.

WSS Weighted Sum Scalarization.

List of Figures

List of Tables

Overview

Chapter 1

Introduction

The computer was born to solve problems
that did not exist before.

Bill Gates

Abstract: This chapter presents a brief introduction to the research work done in developing robust techniques to process visual information, as well as the motivation, the main objectives stated at the beginning, and the methodology employed in this Ph.D. thesis. Several optimization algorithms are improved with robust methodologies based on the use of specific norms. Three different image processing techniques are explored in order to enhance the quality of the state-of-art methods: segmentation, curves, and surface fitting, and super-resolution. In the first case, new segmentation models for natural images in background modeling systems for Pan-Tilt-Zoom (PTZ) cameras and wound identification are proposed by the use of neural networks and ℓ_2-norms. Secondly, improved fitting algorithms for ellipsoids, parabolas, and ellipses using the ℓ_1-norm are presented. Finally, the noise and resolution deficiencies inherently present in medical images are addressed through deep learning techniques and the ℓ_p-norm.

1.1 Motivation

Nowadays, image processing is an active research field. The area progress is boosted continuously with the help of mathematics, computation, as well as better knowledge of specific organs of the human body that intervene in the perception of the images. Advances in digital image processing are reflected in medicine, astronomy, geology, architecture, or surveillance.

Figure 1.1: Examples of a color image with noise (left) and without noise (right) after applying a noise filter algorithm.

Just as our eyes perceive the image through the pupil, the cameras act similarly by emulating this process according to a model known as the "Pinhole camera". In this model, the camera perceives the reflected light rays on the object surface through a small hole known as a pinhole. These rays passing through this hole continue with their trajectory, so a beam of light coming from the upper part of the pinhole reflects in the lower part of the camera and, similarly, rays coming from the left side of the pinhole project onto the right side of it. This process produces a two-dimensional image on the back of the camera, where a sensor or film captures it. As a consequence of the way the light passes through the pinhole, the image is formed upside down. Hence, the digital camera rotates it to give us a familiar representation of the stage or object that we captured. The issue is that one of the main problems that affect image acquisition is the presence of noise and the lack of enough resolution.

The primary sources of noise appear during the acquisition and the transmission of images. There are many types of noise. Among the most known are Gaussian and impulse noise. Several noise removal techniques have been developed, as the filter algorithm. They are a collection of methodologies that, starting from an initial noisy image, aim to obtain a final restored image with enhanced quality, as shown in Figure 1.1. On the other hand, the acquisition procedures often do not allow to generate images with enough resolution, and this makes it necessary to apply an image processing method to augment its resolution. This technique is named as Super-Resolution (SR), which is a broad research field and in continuous evolution.

In particular, in Magnetic Resonance Imaging (MRI) and other medical imaging techniques such as Computerized Tomography (CT), Positron Emission Tomography (PET), or Single Photon Emission Computed Tomography (SPECT), pixel data is affected by several degrading factors, especially finite spatial resolution and noise. The use of Magnetic Resonance (MR) images, both for clinical use and research, remains in continuous growth since its appearance due to its non-invasive character and the possibility of obtaining reliable three-dimensional representations of the structures of the body, increasingly with better resolution, and with the ability to differentiate soft tissues and obtain its functional information.

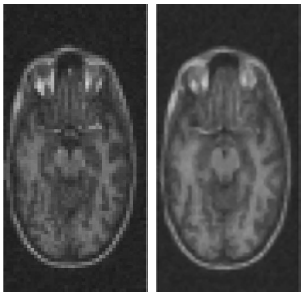


Figure 1.2: Examples of a MR low resolution image (left) and a high resolution one (right) obtained applying a super-resolution algorithm.

The absence of resolution may involve loss of details that could be critical to medical diagnosis. An example of this situation is shown in Figure 1.2. Since acquisition times and protocols are limited for practical reasons, various post-processing algorithms have been proposed to enhance resolution, and are continually being improved. On the other hand, a limitation of this type of images is noise, due to the procedures of acquisition of the image and limited time to acquire it, because of the need for patient comfort, and because some imaging modalities require high-speed acquisitions. Noise greatly affects image quality in some image modalities and generally has a negative effect on the contrast and visibility of details that might contain vital information. An added difficulty in MRI comes from the nature of the noise present in these images, which follows a Rice distribution. Being able to improve such images would provide a breakthrough for medical research and disease diagnosis.

In the field of surveillance, for example, in addition to the previous problems, there are other tasks such as object detection, classification, etcetera, starting from images and videos, which are under investigation due to their vast application potential. Concretely, modeling the background and foreground of a scenario is a complex task due to the presence of many external factors. Being able to develop robust algorithms that contribute, albeit to a lesser extent or as pre-processed, to solve these problems is also part of the present work.

On the other hand, pattern recognition in two-dimensional or three-dimensional data is a widespread problem in many fields of science and engineering, mainly due to the presence of noise. Achieving robust fitting methods against very noisy situations would provide us with a significant advance in image processing under extreme conditions, and allow post-processing algorithms to be much more effective. Typical examples related to this are the fitting of geometrical shapes or the segmentation, which are present in both the real world and medicine. Specifically, in medical imaging, the segmentation of parts of the body with particular characteristics is crucial for further

Figure 1.3: Example of geometric shapes present in the real world.

analysis of the evolution of the patient's disease. Thus, the development of automated object segmentation techniques is vital to facilitate the doctor's task and to extract more accurate and reliable information, which can sometimes escape the sight of the human eye. In addition to this, patterns are always present in many situations, as the examples depicted in Figure 1.3. These also are present inside the body, and providing robust algorithms to fit specific patterns would improve the segmentation procedures.

The above-mentioned problems are only some of the many image processing techniques that exist. Concretely, the most important ones are:

- Feature extraction.
- Pattern recognition.
- Detection.
- Classification.
- Image enhancement.
- Multi-scale signal analysis.

One of the possible workflows in image processing is the sequence formed by the segmentation, pattern recognition, and the enhancement of the image, or more specifically, its resolution. The skull stripping technique is an excellent example of this procedure (Mirajkar et al., 2012). This process of extracts the brain and removes the skull and any other part (as the eyes). For that purpose, the tissue segmentation is necessary to be carried out, and also remove the background. Then, active contour is the typical method to determine the brain mask by fitting an ellipse as the initial contour. Finally, the resolution of the brain image may be enhanced to see finer details.

1.2 Research problem and hypothesis

The main problem this research tries to address is that of developing robust computational intelligence techniques to improve the actual information processing detailed above. State-of-art techniques are very good at tackling specific applications instead of facing the problem with a general perspective, producing solutions that yield good results for specific training data. However, the same methodology can not be applied for similar researches.

Therefore, the problem consists of trying to develop methods that allow us to solve all related problems at once, or being able to generate meta-models that are efficient and accurate regardless of the type of data used. Starting from various classical techniques, such as median filters, interpolation, histograms, adjustment models, etcetera, and also using the power of neural networks, this work aims to take benefit of their advantages and solve problems in several knowledge fields. In particular, neural networks, computationally speaking, are an information processing system whose mechanism of operation is inspired by what is now known from the biological-neurological system. Incorporating some of these latest techniques into our work, such as deep learning, would increment the possibility of achieving more accurate algorithms.

Most of these techniques are based on the optimization of an objective function that tries to minimize some error, which is usually based on a specific norm on a normed vector space. The most common is the well-known Euclidean norm or ℓ_2-norm, which has the intuitive notion of length of the vector. However, although this measure has become a standard when trying to minimize a function, it is not the most robust one.

Thus, we formulate the following research hypotheses:

H1: The use of specific variants of the ℓ_p-norm can improve the performance of the minimization functions.

H2: The ℓ_p-norm can be applied to different computational tasks and increase the robustness of solutions.

1.3 Objectives

Our main objective is to design, implement, and evaluate new robust models and algorithms that allow solving the previous problems mentioned. For that purpose, it is essential to study the basis of the minimization functions as well as the properties of the normed vector spaces we are going to work on. The situations where the optimization systems are going to be applied are so diverse, and they may have different restrictions. Then, designing methods that can be adapted to future needs is an obligation, both generic and specific algorithms, according to specific scenarios.

Thus, the main task will be to eliminate noise or/and enhance the resolution of images. Concretely, the research focuses on three-dimensional medical images. There are not many 3D models developed, and most are based on the use of two-dimensional information from a slice, but that way can lose relevant information about the details of the image.

The segmentation of the images is another pre-processing step that this thesis tries to improve. As mentioned before, this is an essential step to detect objects and measure their features. Although there are several segmentation

techniques in the literature, we will focus on the improvement of some of them.

On the other hand, in both images of nature and those of the human body, we find elements that have the form of known geometric figures (balls, fruits, tumors, brain injuries...) that are corrupted by the presence of noise. This is the step that follows the segmentation in many cases. Adjusting these types of figures to images using minimization methods is a complex problem to be solved. Therefore, one goal will be the development of robust noise adjustment models.

Finally, also a general goal will be the application of neural networks to image processing. Convolutional Neural Networks (CNNs), inspired by the animal visual cortex, have been one of the first deep learning architectures that have demonstrated excellent performance in any field associated with the image and video processing. Thanks to the power of the new graphical acceleration devices (GPU), CNNs have been successfully applied both detection and recognition of objects (Zhang et al., 2016; Cheng et al., 2016), classification of images, or within recommender systems. Hundreds of articles have been published in several areas (Liu et al., 2017; McCann et al., 2017), including the field of medical image analysis (Litjens et al., 2017; Xiang et al., 2017), where the CNNs popularity is growing, and their use are progressively expanding. Developing networks that learn the characteristics of the image to segment, detect, and classify objects, or enhance its quality, fall within the thesis development line.

In order to fulfill these general objectives, the following more specific objectives have been addressed:

O1: Study the existing mathematical backgrounds about optimization functions and robust measures, increasing the existing knowledge about the problem.

O2: Study the recent state-of-art methodologies about pattern recognition and image processing to innovate and improve the current techniques, in particular, those related to super-resolution, curve fitting, and background detection problems,

O3: Study the application of traditional and new measures based on the ℓ_p-norm to reuse algorithms that do not offer adequate performance.

O4: Develop segmentation techniques to improve object detection tasks using neural networks.

O5: Develop optimization methods to fit curves and surfaces to a set of points that are strongly affected by noise or contains anomalous points.

O6: Develop robust super-resolution methods through deep learning techniques for MR images using the volumetric 3D information.

1.4 Methodology

Robust algorithms have to meet several requirements when offering their solution, in addition to solving the problem for which they were designed. The complexity of these methods should not be high in order to solve the problems that require to be implanted in a real-time system. However, in medical imaging, the most important thing is the accuracy of the algorithm regardless of the computation time, since these methods are intended to be used for disease diagnosis. Therefore, the quality of the method will be determined depending on the field of application.

In this sense, the following methodological principles are used for the development of our thesis:

- *Scientific method* (Anand and Cheresh, 2011): This method is the basis of our research. Any project following this methodology will be subject to the principles of reproducibility (to be able to repeat the experiment) and refutability (to be able to reject the hypothesis). It is characterized by following well-defined stages:

  1. Observation: real problems and the existing models are analyzed to build new robust ones in order to solve them.
  2. Hypothesis: we hypothesize that robustness can be improved using minimization function based on the ℓ_p-norm.
  3. Experimentation: a batch of experiments is carried out in order to train and test the proposed algorithms in both toy and real scenarios.
  4. Analysis: the performance of our models is compared with state-of-art implementations.
  5. Conclusions or scientific thesis: the conclusions of the research are extracted to confirm or reject our assumptions.

- *Iterative and incremental methodology*: The development of the project is carried out on the basis of which functionalities will be added, taking into account the highest priority needs at that time, without this addition adversely affecting the rest of the project.

- *Implementation methodology*: An appropriate programming style should be used to make the obtained program modifiable, modular, extensible, simple, documented, etcetera.

- *Evaluation criteria*: the use of adequate performance measurements for each situation. For example: in image restoration, the restored images need to be compared with the original images, which have been used to generate the degraded ones. If we are working with functions, some

specific error measures between the estimated mathematical function and the original one should be defined.

- *Comparison with other models*: consists of studying the differences in results produced by our method from others of state of the art based on the performance measures.

The implementation of these new models is made using various techniques such as optimization methods, statistical models, and neural networks. They will have to fulfill the characteristic of robustness, which is going to be accomplished based on the mathematical norm.

1.5 Thesis contributions

The main contributions of this Ph.D. thesis are the design, study, and implementation of new computational intelligence techniques to address image processing problems. The aim is to make scientific contributions not only to improve specific methods but also to create techniques that can be applied to other fields of research. The contributions can be summarized as follows:

- A review of state of the art for image segmentation, curve, and surface fitting, and image super-resolution.
- The creation of video-surveillance tools to model the background of panoramic images.
- An image segmentation model for heterogeneous skin wounds.
- Three robust geometrical fitting methods against outliers.
- Four SR methods to increase the resolution of 3D images.

1.6 Thesis structure

The structure of this Ph.D. thesis is organized into four parts:

Part I describes the scientific basis for this research. Then, the state of the art in segmentation, point fitting and image enhancement are analyzed in Chapter 2, image concept and MRI are described in Chapter 3, optimization and learning techniques are presented in Chapter 4, and the mathematical frameworks of this research is defined in Chapter 5.

In Part II, two background modeling systems are developed (Chapters 6 and 7), and an additional segmentation algorithm for diabetic wounds is presented (Chapter 8).

In Part III, the geometric curve and surface fitting problem is addressed. Firstly, methods for parabola (Chapter 9) and ellipse (Chapter 10) fitting are

implemented for the two-dimensional plane. Secondly, the 3D generalization of the ellipse is studied, that is, the ellipsoid fitting problem (Chapter 11).

In Part IV SR methods are presented for MRI. An algorithm based on the random shifting technique is developed (Chapter 12), and then the noise is also removed at the same time that the resolution is increased (Chapter 13). By last, ℓ_p-norms are used for the cost function of the SR deep network (Chapters 14 and 15).

Finally, the last block is made up of Chapter 16, which details the general conclusions obtained after completing this thesis and describes the lines of future work.

Part I

Context and Fundamentals

Chapter 2

State of the Art

Real knowledge is to know the extent of one's ignorance.

Confucius

Abstract: This chapter presents a review of the most recent topics that this thesis covers. First, the problem of object detection in video surveillance is analyzed from the perspective of one of the classical pre-processing steps, the determination of the background of a scenario. Also, image segmentation techniques for medical diagnosis are analyzed. Afterward, we sum up the latest works on an ellipse, parabola, and ellipsoid fitting, as well as their applications. To end, the fundamentals of super-resolution are explained, and the classical and the newest techniques are listed.

2.1 Image segmentation applications

2.1.1 Background modeling in video surveillance

The main problem we encounter when addressing the problem of automatic video surveillance is the computational processing of images, where it is necessary to provide intelligence to computers so that they can automatically analyze the desired images. This remote sensing processing can be broken down into three phases: the first stage of detection of moving objects (in the foreground), a second phase that controls the tracking of those objects, and a third and final action consisting of behavioral analysis.

This whole automatic process is quite complex. At this point, our task is to focus on the first phase, where it is critical to model the background in order to increase the efficiency detecting the foreground objects that appear

in them. Depending on the scenario, this task may be more or less difficult due to the numerous inconveniences that affect the development of this task (Bouwmans et al., 2018). Some of the most important are the following:

- *Scene lighting and shadows*: when the stage lighting changes, the background that the algorithms model contains different shades from the current scenario, which will affect those parts with different hues being recognized as foreground. Besides, a change in lighting can cause shadows.
- *Dynamic backgrounds*: scenarios where a repetitive movement of one element is present, such water of a river or the branches of a tree.
- *Moving background objects and static foreground objects*: a punctual movement of an object that is considered from the background, or a foreground object stays still for quite some time, can perturb the learned model.
- *Noise*: the video may contain noise as a result of a faulty engraving process or a poor quality device.
- *Camera movement*: a static camera can suffer momentary vibrations and the following error in the modeling of the background. For example, the incidence of the wind produces a movement of the recorded scene by the camera.

These factors make it essential to develop robust models to separate the background adequately from the foreground (Jeeva and Sivabalakrishnan, 2015). Many researchers have proposed robust methodologies reconstruct the background based on distance measures (Yang et al., 2016), or the use of multiple cameras (Cermeño et al., 2018). However, a determinant context to be considered in the modeling is the type of camera used in each situation.

Most cameras used in surveillance are static, and the scenes taken from this type of cameras only show one specific view of the surveillance area. For the images/videos captured by static cameras, the most common and efficient approach to moving object detection is background subtraction, which consists of maintaining an up-to-date model of the fixed background and detecting moving objects as those that deviate from such model (Cheung and Kamath, 2004). Compared to other approaches, such as optical flow, this approach is computationally affordable for real-time applications, is independent of moving object velocity, and is not subject to the foreground aperture problem.

However, traditional background subtraction algorithms assume the cameras are static, and this leads to false detection when the camera moves (Kim et al., 2013; Xue et al., 2013). Due to this camera movement, even pixels belonging to static objects appear to move in the camera frame (called the

ego-motion effect). Extensive research has been carried out regarding object detection for moving cameras. Some proposals are based on the optical flow clustering, which consists in calculating dense or sparse optical flows and clustering them to identify moving object regions (Varcheie and Bilodeau, 2011). Other methods are based on the estimation of the transformation parameters between consecutive frames (López-Rubio and López-Rubio, 2015).

The approaches presented in our research are based on mosaicing the background (Azzari and Bevilacqua, 2006; Bevilacqua and Azzari, 2006), which consists of creating a mosaiced or panoramic background image and then using a background subtraction technique to extract moving object regions.

2.1.2 Body lesion pre-processing

Most of the automatic diagnosis systems in medical imaging have suffered the problem of data availability, causing an insufficient capability of generalization of the prediction models. In addition to this, training datasets lack sufficient quality in the sense of homogeneity in the acquisition procedure and non-expected objects present in the image, making it necessary to carry out several pre-processing steps (Asha Gnana Priya et al., 2018) and segment the region of interest (Jafari et al., 2016, 2017). Moreover, another commonly used technique is the extraction of features that are used then to improve the accuracy (Devassy et al., 2019; Yadav and Kaushik, 2018).

The main inconvenience of these approaches is the requirement of specific expertise to extract the adequate features and the high quantity of time necessary to select the most appropriate ones. Moreover, image pre-processing may introduce errors or loss of essential information that can affect the final performance. A simple example is the low accuracy obtained when a poor segmentation of the lesion is carried out. Until a few years, the classical workflow was the use of these traditional techniques (Oliveira et al., 2018), yielding not good enough accuracy. In order to overcome these limitations, deep learning models have recently been developed with success, having the ability to automatically learn the crucial features that allow us to perform accurate image pre-processing.

Deep learning has been applied to resolve very complex classification and segmentation tasks (Pereira Dos Santos and Antonelli Ponti, 2019; Shahin et al., 2019) without the use of any image pre-processing method. The architecture of these networks is mostly based on convolutional layers, which filter and extract essential image features in order to learn the different lesions. For example, Zhou et al. (2019) use different modality images to learn the features that determine dementia cases. Nowadays, with the increase of publicly available datasets, the deep networks have shown an excellent performance in medical image analysis (Litjens et al., 2017). Gao et al. (2019) used neural networks fed with extra privileged information to carry

out strain reconstruction in ultrasound elastography. Deep learning models have also used to detect vessel borders (Gao et al., 2020a) and perceive blood flow from angiographies (Gao et al., 2020b). Specifically, recent works for skin lesion classification (Nida et al., 2019) have been published, although still there is a margin of improvement. These works are based on a two-stage process, so they can segment and extract features with deep networks and then make the prediction.

This research focuses on the segmentation step. Many classical techniques have been developed with success (Pal and Pal, 1993). For instance, Comaniciu and Meer (1997) presents a color-based segmentation based on the mean shift algorithm, being very robust in extracting the significant features of the image. Another flawless application is the whole brain segmentation (Fischl et al., 2002) or atlas-segmentation (Lötjönen et al., 2010) in magnetic resonance, where the precision has to be very high. Patch-based segmentation is also heavily used in this field because of the similarity present in other parts of the brain (Coupé et al., 2011). Providing a robust system to segment lesions with deep networks still is a challenge due to the lack of enough data and its heterogeneity.

2.2 Geometric figures fitting

In recent years, research concerning computer vision has developed rapidly. Fitting primitive models such as a quadratic curve or a surface to a given cloud of points is a fundamental task in digital image analysis, visual pattern recognition, computer vision, and computer graphics with numerous applications. The 2D case involves fitting a quadratic curve to a set of sample points on the plane (Chernov and Lesort, 2005; Ladrón De Guevara et al., 2011; Szpak et al., 2015; Muñoz-Pérez et al., 2014; Waibel et al., 2015). Also, the ellipticity and circularity of the set of samples can be measured (Misztal and Tabor, 2016). The 3D case consists of fitting a quadratic surface to a set of points in space (Sahin and Unel, 2008), which has applications to 3D reconstruction (Cross and Zisserman, 1998), pose estimation (Lee and Moore, 2004), restricted stereo correspondence problem (Collings et al., 2009) and object recognition (Cao and Shrikhande, 2002), (Sappa and Rouhani, 2009). Moreover, fitting algebraic surface with scatter 3D points has been discussed widely in (Li and Griffiths, 2004) and some remarkable work has been done in (Keren et al., 1994; Keren and Gotsman, 1999; Pratt, 1987; Sullivan et al., 1994; Taubin, 1991; Kriegman and Ponce, 1994).

Given a parametric function

$$P(x, g) = 0 \tag{2.1}$$

where P represents a quadratic function, x stands for the point coordinate vector, and g denotes the curve or surface parameters, the main purpose is

to estimate g from noisy samples

$$x_i = \bar{x}_i + \tilde{x}_i \tag{2.2}$$

where $\bar{x}_i$ is noiseless data and $\tilde{x}_i$ is a noise contribution such that for all $\bar{x}_i$ we have

$$P(\bar{x}_i, g) = 0 \tag{2.3}$$

One of the most accurate approaches to estimate g in Eq. (2.1) is with maximum likelihood (ML) methods, where the geometric distance of data points to the curve or surface P is minimized. These methods are typically computationally expensive and usually converge slowly if the minimization is formulated as an iterative process. Traditional least squares methods (LSM), in contrast, which minimizes the squared equation error, also called as algebraic fit, are cheaper and involve no iterations but may be heavily biased and non-robust systems (Chernov and Lesort, 2004; Kanatani, 2008).

The main objective of (Hunyadi and Vajk, 2014) is to use the simplicity of the algebraic methods (Eq. 2.1), avoiding the statistical inaccuracy in least-squares approaches (Eq. 2.2), while enforcing constraints that arise from a priori information at a low computational cost (Eq. 2.3). They work with images where data points are polluted by a large amount of Gaussian noise.

Depending on the type of curves or surfaces, the definition of the quadratic function P and the minimization techniques used to reach the convergence of the fit may vary. Next, three geometrical figures are analyzed, parabolas, ellipses, and ellipsoids.

2.2.1 Parabolas

As it has been mentioned before, for the conic fitting, there are two problems to be considered (Li et al., 2012). One is the mathematical form of the formula for the fitting conic, and the other is the objective function of the fitting conic. The selection of the objective function, during the fitting of the curve, can be divided into two categories: firstly, based on minimizing the geometric distance from the point to the curve and secondly, based on minimizing the algebraic distance.

The general form of a conic section is:

$$Q(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{2.4}$$

The roots of Eq. (2.4) evaluated at infinity allow to identify the type of conic (Harker et al., 2008).

Explicitly,

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{2.5}$$

Through (2.5), the type of conics can be characterized as follows:

$$B^2 - 4AC \begin{cases} > 0 \, real \, asymptotes \Rightarrow hyperbola \\ = 0 \, real \, parrallel \, asymptotes \Rightarrow parabola \\ < 0 \, complex \, asymptotes \Rightarrow ellipse \end{cases} \tag{2.6}$$

The discriminant $B^2 - 4AC = 0$ depicts a second-order surface in the (A, B, C) tridimensional parameter space, i.e., it is an elliptical cone with the origin as its vertex. This surface represents all parabolas. All ellipses are enclosed inside the cone, whereas all hyperbolas are external to it.

According to Bookstein (Bookstein, 1979), the invariance must be with respect to transformations of the Euclidean plane, such as rotations, translations, and scale changes. If the dataset coordinates are transformed, then the resulting best fitting conic should match the best fitting conic that would be obtained by the algorithm if the non-transformed samples were provided to the algorithm. The Yu method (Yu et al., 2009) has a cost function with a geometric interpretation where the parameters are intrinsic of the conic and are translation and rotation invariant.

In recent years, there are some research works in the medical field. To cite an instance, a study on teeth (Muhamad et al., 2015). Another example in this field is related to the corneal shape. A direct mathematical approach for obtaining relevant parameters of the corneal surface was proposed in (Jannuts et al., 2015). Another study connected to eyelid location using image focus has been carried out by Jang et al. (2008). According to the anatomic research, the parabola-like vascular geometry model is the most common way used to detect fovea (Chin et al., 2013; Yu et al., 2014b,a). Other medical applications that have been studied are presented by Ogul et al. (2015) who intended to solve the segmenting ribs problem in a chest radiography image as a halfway step for eliminating rib shadows for an effective Computer-Aided Diagnosis System (CAD). The proposed system facilitated a novel strategy to fit a parabolic curve to all rib seeds acquired by a log Gabor filtering approach. In the same problem as image diagnosis for prostate cancer recognition, Samarasinghe et al. (2015) proposed a semi-quantitative model to represent perfusion behavior of 3-dimensional prostate voxels in DCE-MRI sequences based on parametric evaluation of parabolic polynomials. Perfusion data of each prostate voxel is modeled on to a best fit parabolic function using second-order non-linear regression.

Moreover, telecommunications and industrial applications are presented by Kampanis et al. (2013); Man and Li (2014); Späth (2001); Wang et al. (2014); Yan et al. (2007). In (Späth, 2001), the problem of fitting a rotated paraboloid to given measured data in 3-space has been discussed. Furthermore, Wang et al. (2014) claimed that one of the most important issues for engineers implicated in the structural design of a large antenna is compensation for degrading electromagnetic efficiency. Other industrial applications

have been carried out by Yan et al. (2007). They integrated the analogy of gray value of power lines into particle filtering to track the points on power lines, and use those selected points to fit the power line as a parabola.

On the other hand, the automated detection of lanes is an essential part of driver assistance systems in smart vehicles (Narote et al., 2018; Niu et al., 2016). For example, Wang et al. (2010) proposed a new kind of lane boundary detection algorithm based on a parabola model in order to improve lane detection accuracy under different road conditions for intelligent vehicles. Another research presented by Li et al. (2016) showed a multilane detection method based on omnidirectional images to address the difficulties from the limited view field of the rectilinear cameras. In the physics field as detecting in (Man and Li, 2014) was presented an accurate measurement method for optics system with lunar imaging. The results show that the method is highly accurate for focal length measurement.

An atmospheric application related to sound propagation can be found in (Kampanis et al., 2013). Parabolic equations, approximations of the Helmholtz equation in cylindrical coordinates, are used widely in the underwater acoustics context, see (Dougalis et al., 2008; Lee and McDaniel, 1987). They have been further employed to simulate the analog problem of sound propagation in an inhomogeneous atmosphere. Finite element (Gilbert and White, 1989; Raspet and Di, 1990), finite difference (Robertson et al., 1995, 1996), and Fourier/Green's function methods (Gilbert, 1993; Salomons, 1998), have been used to discretize the parabolic equations in cylindrically symmetric domains. In parabolic equation models, a waveguide with irregular boundaries can be encountered. In computational fluid dynamics (Magoulès, 2011), a common practice to transform the physical domain to a rectangular computational one, is the use of a boundary fitting curvilinear coordinate system. In (Dougalis and Kampanis, 1996), a similar analytical transformation of coordinates was used in a finite element discretization of the conventional parabolic equation in a sea environment with variable bottom. Irregular terrain is also encountered in electromagnetic parabolic equation models (Teti, 2001).

Finally, an example in the architectural field is showed in (Samper et al., 2017), which provides a method that expects to determine the best fits to the geometric shape of an arch of a heritage building. This method only involves standard geometric processes, computing, statistics, numerical processes, and data acquisition.

2.2.2 Ellipses

The condition $B^2 - 4AC < 0$ must be fulfilled in order that the general equation (2.4) defines an ellipse. Like in the parabola case, fitting methods can be divided into algebraic and geometric fitting. If the value of a point (x, y) is replaced in the equation, the value obtained in the implicit equation

is called an algebraic distance. Also, the solution is not always an ellipse in spite of the algorithms being efficient. On the other hand, geometric fitting tries to minimize the geometric distance, i.e., Ahn et al. (2001), who minimize the sum of the orthogonal distances.

Many researchers have tried to obtain accurate and robust methods. They approximate the geometric distance by a function of the ellipse parameters. Yu et al. (2012) have determined a geometric objective function considering that the sum of the distances from a point to the foci is constant. On the other hand, Kanatani and Rangarajan (2011) have presented an algebraic method using the implicit ellipse equation with a stipulated constraint. This gives rise to an accurate algebraic solution by full error analysis. However, parabolas and hyperbolas are described by the general equation of the conic as a solution and the sensitivity to outliers. Liang et al. (2019) used the alternating direction method of multipliers on an adaptation of direct least-square fitting using the ℓ_p-norm with $p < 2$.

In an algebraic fitting, the curve is given by a constrained implicit equation of a conic. This fitting has implementation and computing costs advantages (Muñoz-Pérez et al., 2014), but also some drawbacks in accuracy, physical interpretation of the fitting parameters, errors, and sensitivity to outliers. Although the algorithms are efficient, the solution is not always an ellipse.

Nevertheless, several kinds of research have been working on least-squares problems based upon the square of the sum of algebraic distances or its variations, (Fitzgibbon and Fisher, 2013; Kanatani, 1994; Sampson, 1982). As reported by Ahn et al. (2001), some fit drawbacks have been resolved by other authors. Therefore, the Direct Least Square method was one of the significant advances in algebraic procedures suggested by Fitzgibbon and Fisher (2013). A new computationally efficient constraint was their contribution, which guaranteed that an ellipse was the optimal solution. On the other hand, Ahn et al. (2001) used the Orthogonal Least Squares Fitting, introducing some enhancements which overcame the weak points of this fitting scheme. They try to minimize the sum of the orthogonal distances. This criterion has a strict geometric interpretation because the Euclidean distance from the points is used as an error measure to solve the issue. However, it must be solved iteratively.

The geometric distance is employed by many researchers using a function of elliptical parameters; in other words, the "Sampson error" (Muñoz-Pérez et al., 2014; Sampson, 1982). Kanatani (1994) proposed a renormalization, Chojnacki et al. (2000) a Fundamental Numerical Scheme (FNS) or Leedan and Meer (2000) and Matei and Meer (2006) Heteroscedastic Errors in Variable (HEIV). Kanatani and Sugaya (2010) have proved that the Sampson error shows an excellent estimation of the geometric distance, and its minimization outcome is close to the true geometric fit. Meanwhile, Calafiore

(2002) presents a fitting solution for a set of points in reference to the model of the difference of squares geometric error. The proposed algorithms are based on a closed-form solution that guarantees a global minimum is reached in a limited amount of iterations.

Genetic algorithms have been used by De La Fraga et al. (2007) and Ray and Srivastava (2008) to solve optimization problems of ellipse fitting. The purpose is to minimize the sum of orthogonal Euclidean distances from the given points. Roth and Levine (1993) applied the Least Median of Squares as a robust estimator, and it has been contrasted to other robust processes such as Rosin (1999). On the other hand, Yu et al. (2012) determined a new geometric objective function considering that the sum of the distances from that point to the foci is constant. Finally, Muñoz-Pérez et al. (2014) used the criteria in reliance on the least mean absolute geometric error considering that the optimum value of the sum of distances from the points to the foci is computed by using the median, a robust estimator. This method detects the presence of outliers (Meer et al., 1991). So, other methods like RANSAC (Calafiore, 2000) shown by Fischler and Bolles were not necessary.

Some illustrations of the importance of the ellipse fitting methods have been researched. One example is the eye localization problem, where it is needed to face recognition, device interaction, face alignment, etcetera. Regarding industrial environments, another issue is a camera calibration based on the ellipses fitting due to the projection of cylinders that are used to determine the camera position and orientation. In other application fields such as biology, chemistry, and nanotechnology, ellipses fitting is used. Li (2019) shows a reliable, effective, and accurate approach to this type of problem and, in a particular one, a handprint identification issue.

As an example of the variety of applications, Islam et al. (2020) introduce an ellipse fitting method in non-invasive procedures used in vascular permeability images, relevant for monitoring cancer solid tumors based on the use of ultrasound poroelastography. Borges (2017) presents a geometric fitting method based on the least-squares. This method is used to analyze rocks' properties and other materials. Mitchell and Van den Berg (2016) use ellipse fitting to analyze electron diffraction in order to obtain different patterns from polycrystalline materials. And Liao et al. (2017) use a method based on detection techniques and ellipse fitting for cell image segmentation.

2.2.3 Ellipsoids

According to Malyugina et al. (2014), the existing methods of ellipsoidal fitting are mainly based on the least-squares method. Methods of this group use an algebraic approach regarding an implicit representation of the quadratic surface that (Li and Griffiths, 2004) defines as the locus of points such that their coordinates satisfy the most general equation of the second degree in

three dimensions:

$$ax^2 + by^2 + cz^2 + 2fyz + 2gxz + 2hxy + 2px + 2qy + 2rz + d = 0 \quad (2.7)$$

Now, let

$$I = a + b + c$$

$$J = ab + bc + ac - f^2 - g^2 - h^2$$

$$K = \begin{vmatrix} a & h & g \\ h & g & f \\ g & f & c \end{vmatrix}$$

then it is known that I, J and K are invariant under rotation and translation and equation (2.7) represents an ellipsoid if $J > 0$, $IK > 0$ in (Harris and Stocker, 1998).

In literature, a wide range of fitting techniques has been applied depending on the researcher. There are two main categories of fitting methods (Ying et al., 2012; Malyugina et al., 2014), namely using algebraic distances (Blane, 2000; Helzer et al., 2004) or geometric distances (Sung Joon Ahn et al., 2002; Chen and Liu, 1999; Gander et al., 1994; Gotardo et al., 2004; Sappa and Rouhani, 2009; Sturm and Gargallo, 2007) according to their error distance definitions. More details of these references are found in Chapter 11. Algebraic fitting methods often yield a unique noniterative solution, while geometric ones make use of some non-linear constraints by iterative algorithms (Muñoz-Pérez et al., 2014).

In 2004, Li and Griffiths (2004) explain that ellipsoid fitting can be accomplished in many ways by applying some known bounded surface fitting techniques, see (Keren et al., 1994) and (Kriegman and Ponce, 1994). Even so, these techniques involve a non-linear optimization method, which cannot guarantee an optimal solution and stops at a local minimum. Their purpose is to develop an effective and efficient ellipsoid fitting algorithm and not to consider those fitting methods based on the geometric distance or the parametric representation of an ellipsoid, since they give rise to a lengthy non-linear optimization process.

In contrast, Yu et al. (2009) mention that general ellipsoids do not have a natural geometric definition similar to ellipses or spheroids. Regardless, they can still generalize the ellipse fit algorithm to three dimensions in the case of a spheroid, since a spheroid has the same basic geometric property as an ellipse. This is the reason that all algorithms proposed in two dimensions can be easily generalized to three dimensions in a natural way.

The ellipsoid is one of the most used geometric primitives, and has paramount importance in many fields. For instance, it is employed in calibration for three-axis magnetic sensors, as shown in (Li et al., 2018a), where a mathematical model is proposed for scalar calibration of a single three-axis magnetometer and a least-squares ellipsoid fitting algorithm to estimate the

detailed error parameters. Three-axis magnetometers are used to measure the intensity of the Earth magnetic field, being widely applied in navigation systems and aircraft (Malyugina et al., 2014). According to Gietzelt et al. (2013), it is known that calibration algorithms for three-dimensional accelerometers and magnetometers are equivalent to a 3D-ellipsoid fitting problem. Most calibration algorithms are based in 3D ellipsoid fitting methods, which minimize a cost function to estimate the value of a set of calibration parameters that define the sensor model. These algorithms are generally employed in micromagnetometer calibration for accurate orientation estimation (Zhang and Yang, 2015), micromagnetometers, together with inertial sensors and also, to calibrate Magnetic Angular Rate and Gravity (MARG) sensors (Olivares et al., 2013), or in the integrated calibration of magnetic gradient sensor systems (Li et al., 2018a).

Many commercial electronic devices need to solve 3D ellipsoid fitting problems. Smartphones, tablets, GPS navigators (Feng et al., 2013; Lou et al., 2015), watches and videogame peripherals comprise accelerometers and magnetometers that are used to estimate the orientation of a body in space (Fang and Liu, 2015), compute trajectories (Krach and Robertson, 2008; Skog et al., 2010; Park and Suh, 2010), analyze motion (Susi et al., 2013; González-Villanueva et al., 2013; Chung and Ng, 2012), detect obstacle collision (Rimon and Boyd, 1997), and other innovative measurement applications such as 3D scanning (Grivon et al., 2013) or in reverse engineering by a laser scanner point data from the surface of an object (Lukács et al., 1998).

In the medical field, fitting ellipsoids have been used for estimating diameters of pulmonary nodules (Kubota and Okada, 2005) as a technique to extract a pulmonary nodule from helical thoracic Computerized Tomography (CT) scans and estimate its diameter. Another outstanding study is presented by Ge et al. (2005), in which a computer-aided detection system is developed to assist radiologists in the detection of lung nodules on thoracic CT images, using a 3D gradient field method and 3D ellipsoid fitting. Furthermore, Karnesky et al. (2007) has studied the coalescence of γ' ($L1_2$) precipitates in Ni–Al–Cr through the best-fit ellipsoids of atom-probe tomographic data, since best-fit ellipsoids have equivalent centroids, moments of inertia, and principal axes for arbitrarily shaped precipitates. Additionally, Mahdavi and Salcudean (2008) suggested a 3D semi-automatic prostate segmentation method for B-mode Trans-Rectal UltraSound (TRUS) images, where it is assumed that the prostate transversal section shape may be a tapered ellipse.

Other research fields are focused on body detection or classification, such as nose tip detection by Sarakon et al. (2013). This is useful because the nose is a facial feature whose shape is not affected by facial expressions and is visible for almost any orientation of the head (Theekapun et al., 2006). Further-

more, the nose is often used as a baseline to localize other facial landmarks such as the eyes and the mouth. Other approaches include Sarakon et al. (2014), which proposes a non-contact method to classify the face shape by using Support Vector Machine (SVM) technique and Mahalanobis distance applied to define a precise head region. Moreover, Grammalidis and Strintzis (2000) suggested a head detection system of a person and tracking system by 2D and 3D ellipsoid fitting by K-means algorithm in head segmentation and 3D ellipsoid model.

2.3 Image quality enhancement

Despite advances in acquisition techniques such as CT, PET, MRI or combined modalities (e.g., SPECT/CT), factors like the inherent noise in the devices or blurring, are always present in each of these techniques, which sets limits to the resolution and quality of the obtained images. MRI is no exception to that, with a resolution of the order of millimeters, due to the magnetic properties of the tissues and varying as a function of the way the signal and noise are sampled and filtered. Hardware limitations, high Signal-to-Noise Ratio (SNR), practical limits to the acquisition time, and patient movement also contribute to limiting image resolution.

The improvement of image quality and resolution is a constant aim in medical imaging, due to the critical importance of these images to find the correct diagnosis and treatment for patients. This is not only reflected in the optimization of acquisition techniques, for instance, in the case of magnetic resonance imaging (MRI), but also at the post-processing stage, with an ever-increasing interest in new, improved algorithms. Enhancing resolution is particularly relevant in this area, given the need to inspect the details of anatomical structures and to locate functional information in a more precise way. According to this need, in recent years, there has been a growing interest in improving super-resolution methods for MRI, linked to the development of machine learning algorithms.

Interpolation and spline-based techniques are the most classic and well-known strategy to upsample MR images (Pan et al., 2012; Lehmann et al., 1999; Thevenaz et al., 2000). Linear and bicubic interpolation methods have been extensively used. However, these approaches estimate new intermediate points assuming the homogeneity of a region, causing a blurring perception in the image, especially in areas with edges or contrast changes. Moreover, these techniques may also introduce ringing and aliasing artifacts.

Super-Resolution (SR) techniques offer a better approach for resolution enhancement (Wang et al., 2018; Tian and Ma, 2011; Van Reeth et al., 2012; Yue et al., 2016; Yang et al., 2014). They can be grouped into two categories, namely, Single-Image Super-Resolution (SISR) and Multi-Image Super-Resolution (MISR).

SISR methods apply SR on a single Low-Resolution (LR) image, which is typically an ill-posed inverse problem. Successful proposals, such as the example-based methods, have become popular as super-resolution techniques (Shah and Gupta, 2012; Balure and Kini, 2015; Prince et al., 2019). Among them, some exploit the internal similarities of the image (Manjón et al., 2010b; Zheng et al., 2017) and others learn mapping patterns between LR and High-Resolution (HR) images from external datasets (Rueda et al., 2013; Trinh et al., 2014; Zhang et al., 2015; Jia et al., 2014). The most frequent single-image approach has been the use of a database of example images, from which the algorithms learn how to estimate the lost HR information in the LR image. In this approach, the images are considered as a set of image patches, and SR is performed for each patch. Many state-of-the-art methods have been proposed, some using nearest neighbor search (Tu et al., 2016; Freeman et al., 2002) and others, more recently, applying a sparse representation approach (Dong et al., 2011; Trinh et al., 2014; Yang et al., 2010). One of the issues of example-based SR is that it highly depends on the database of LR and HR patch pairs. It is also difficult to apply these methods with arbitrary zoom factors since it would be necessary to retrain the model for each new zoom factor. Self-similarity based methods avoid the use of an external database by taking into account that most natural images contain large amounts of repetitive visual information. This allows achieving SR by searching for similar patches in the same image, as done in (Rueda et al., 2013; Jia et al., 2016; Zhao et al., 2019), where the patches are also searched for across different scales.

MISR techniques are based on the acquisition of multiple low-resolution (LR) images of the same object with sub-pixel shifts between them (Farsiu et al., 2004b; Wang et al., 2018; Patil et al., 2018). These LR images need to be registered, fused, and deblurred. Also, the registration step is difficult to apply with the necessary precision, since it is hard to estimate motions between multiple blurred and noisy LR images, even more, when it is necessary to assess the movements of the patient at the same time. Moreover, repeating the imaging process would lead to excessive acquisition times in most medical imaging modalities. These types of techniques are not considered within the scope of this thesis.

Many machine-learning approaches are being proposed for medical imaging applications, and deep learning is becoming increasingly popular among them (Pham et al., 2019). The interest of these algorithms for medical imaging is developing as they evolve towards greater efficiency and reliability. The improvement of image resolution is a fundamental step towards attaining an adequate performance in subsequent phases of the medical image processing pipeline. For example, segmentation of the regions of the human brain by clustering is a key task (Mirzaei and Adeli, 2019), which can benefit from the enhancement of the quality of the input image. Deep learning-based

MRI super-resolution technology has the potential to become a standard procedure in all MRI medical protocols (Chong, 2020).

Recently, a super-resolution convolutional neural network (SRCNN) (Dong et al., 2016), which can be considered as an example-based method, and its 3D version 3D Super-Resolution Convolutional Neural Network (SRCNN3D) (Pham et al., 2017), have obtained great attention because of their ability to learn an end-to-end mapping between LR and HR images, thus avoiding to learn from dictionaries or manifolds to model the high-resolution space. Other recent works also incorporate neural architectures based on deep learning to perform super-resolution tasks (Liang et al., 2016; Li et al., 2017; Lin et al., 2018; Chang et al., 2018; Li et al., 2017).

A 3D densely connected super-resolution network for brain MRI data improvement was recently presented (Chen et al., 2018c). Its particularity is the use of dense layers instead of convolutional layers on its architecture. The expected high complexity of the method is mitigated because it is a light-weight model with weight sharing and a very reduced number of parameters. Another recent work presents a combination of a densely connected network with a Generative Adversarial Network (GAN), which seems to achieve promising results (Chen et al., 2018b). Additionally, other exciting works that relate brain MRI and deep learning architectures can be found in the literature (Shi et al., 2018; Park et al., 2018; Xiaole et al., 2019).

Chapter 3

Image fundamentals

Use a picture. It's worth a thousand words.

Tess Flanders

Abstract: This chapter intends to introduce the fundamentals of the different kinds of images used in this research. Therefore, we start by explaining how an image is represented in the computer, and what the characteristics of the two most-known image representations are: gray-scale and color images. Finally, as the most important application of this research is medical imaging, and more precisely, magnetic resonance images, its acquisition procedure is described, as well as how this kind of image is stored digitally and the limitations we can encounter.

3.1 Traditional images

An image is a visual representation, which manifests the visual appearance of a real or artificial object. The encoding of an image for its computer processing begins with its digitization, which involves replacing the actual image (which is often a continuous representation) with a finite set of values that can be handled by computer means. This finite set of values represents the image. Two primary forms of representation are distinguished by the way the digitized image is constructed, bitmap representation, and vector representation.

The bitmap representation of an image is a two-dimensional grid of values located in cells, better known as pixels, that are generated at regular intervals. From here, the pixel set represents the infinite points of an ideal

image. Then, the color points are reduced to numbers for storage and treatment in the grid. Images can be represented on various color scales. The most common are gray-scale and Red-Green-Blue (RGB).

Thus, an image can be represented by a $N \times M$ matrix, and depending on the quantity of information we want to store, more dimensions are employed. Next are presented two classical bitmap image representations: gray-scale and color images.

3.1.1 Gray-scale image

Gray-scale images are the most straightforward representation of an image since it only contains information about the lightness of each pixel. The contrast ranges from black at the weakest lightness (0%) to white at the strongest (100%). These values are usually ranged between 0 and 255 (8-bit precision), although more bits or even floating numbers can also be used if more precision is needed. In order to reduce the required storage and facilitate the computation, image pixels are usually re-scaled to store them as unsigned integers.

3.1.2 Color image

The chromatic information of an image can also be stored within the pixels by using one of the well-known color models, Red-Green-Blue (RGB). This system is a color model based on additive synthesis, with which it is possible to represent a color mix by adding the three primary colors. The RGB color model does not in itself define exactly what red, green, or blue mean, so the same RGB values can display noticeably different colors on different devices that use this color model. Even if they use the same color model, their color spaces can vary considerably.

In order to indicate the proportion of each color, a value is assigned to each primary color. The value 0 means that it is not involved in the mixture, and, as that value increases, it brings more intensity to the mixture. Although the range of values could be any, it is typical for each primary color to be encoded with 8 bits. Thus, as the gray-scale color space, the intensity components are measured on a scale ranging from 0 to 255.

3.2 Magnetic Resonance images

The interest of Magnetic Resonance Imaging (MRI) in medicine has been growing since its inception, for the ability to distinguish organs, tissues, bones, and other structures within the body, and to provide both anatomical and functional information that helps to monitor metabolic processes therein, as well as being non-invasive because it does not involve irradiation of the

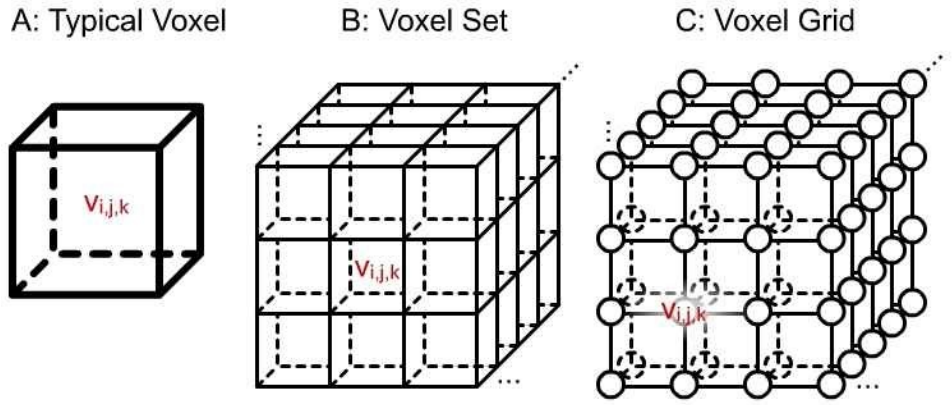


Figure 3.1: Illustration of the voxel.

patient. MRI uses strong magnetic fields and radio waves to produce internal images of the body. Unlike X-rays and CT scans, MRI machines cross-produce 3D images of the body without using radiation (Brown et al., 2014).

The image formats above-mentioned are the classical representation of a bidimensional image. However, in medical tasks, it is interesting to analyze the three-dimensional distribution of the different parts of the body. Thus, this needs an alternative for the usual pixel representation. The voxel (Volumetric Picture Element) is a cubic unit that makes up a three-dimensional object (see Figure 3.1. It is the minimum actionable unit of a 3D array and is, therefore, the pixel equivalent in a 2D object. Like pixels, voxels do not contain their position (x, y, z) in 3D space, but are inferred by the position of the voxel within the data file, and are of a specific size.

In MR images, each voxel will contain a single value, which will be represented by a unit of measurement established during the creation of the images. Therefore, we are dealing with a kind of 3D gray-scale image, but with more precision than usual since it contains fine details obtained during the acquisition procedure. This information gives different intensity values to each voxel. The interior details of an image are usually visualized through slices, so the different intensity values make it possible to distinguish different types of tissues.

3.2.1 Image acquisition

MRI had its beginnings in 1973 when the first publications of Lauterbur and Mansfield came out (Lauterbur, 1973; Lauterbur et al., 2019). The main idea of the principles of magnetic resonance is that the patient is placed in a uniform B_0 magnetic field, which is generated by a superconducting magnet situated around the patient. Also, it is known that the intrinsic

angular momentum (more frequently called 'spin') of a hydrogen nucleus in a magnetic field precesses about that field at the Larmor frequency, which depends linearly on the magnitude of the field itself:

$$\omega_0 = \gamma B_0 \tag{3.1}$$

where γ is a constant called the gyromagnetic ratio. So the body spins have the Larmor frequency that (3.1) says, and MRI is based on the signal from the protons given the abundance of hydrogen nuclei in the body.

If a radio-frequency pulse is sent at the Larmor frequency, the proton spins will enter into an excited state by the MRI effect. When the pulse vanishes, they will return to the unexcited state, in turn emitting a radio-frequency pulse detectable from the outside. The MRI image is formed from this signal. The fact that, depending on the tissue of the body in question, there is a slightly different local magnetic field, makes the signal produced at that point reflect the local tissue, or at least its magnetic properties. However, we not only need to detect a signal that reflects tissue characteristics but also to distinguish the point of the object from which the signal comes. Although the physics behind this technique does not fall into the scope of this research, it is interesting to introduce how the different types of images are formed, in order to know what kind of information they store.

Let us consider the longitudinal equilibrium magnetization M_0 for the component of the magnetic moment vector along the external field direction. For a sample with ρ_0 defined as the number of protons per unit volume, i.e., the spin density, the longitudinal equilibrium magnetization is given by the proton magnetic moment component $\gamma h/2$ multiplied by the relative spin excess times the spin density.

$$M_0 = \frac{\rho_0 \gamma^2 h^2}{4kT} B_0 \tag{3.2}$$

where h is the Planck quantum constant, k is the Boltzmann constant, and T is the temperature, which is practically constant in the body.

Suppose now that $\overrightarrow{M}$ has been rotated by an radio-frequency pulse to a direction orthogonal to $\overrightarrow{B_0} = B_0 \overrightarrow{z}$. A radio-frequency pulse with a frequency equal to the Larmor frequency can achieve this. The new transverse magnetization has magnitude M_0 and begins to precess clockwise in the xy plane. The complex magnetization is defined in terms of the magnitude and the polar angle phase, which gives the direction in the xy plane of the 2D transverse magnetization, as

$$M(t) = M_0 e^{-i\omega_0 t + i\phi_0} \tag{3.3}$$

The goal of MRI is to correlate all the signal measurements with the spatial locations. Each chemical species has its contribution to the total

signal. Nevertheless, when all protons are represented by only one species, like water, and considering that the field is uniform, for example, then the total signal from all spins is given regardless of their spatial location in the static magnetic field.

If a linear gradient is applied to the B_0 field in a certain direction, the Larmor Frequency will also vary linearly in that direction:

$$\omega(x) = \gamma B(x) \tag{3.4}$$

where x denotes the spatial coordinate along the direction of the gradient of the field. A radio-frequency pulse at a certain frequency ω will only be able to excite spins with a Larmor frequency equal to ω, therefore only those with a cut in a direction perpendicular to the gradient. We can then measure the signal of that cut separately from the others. This means that the spectral components now represent spatial information, and the object can be reconstructed to form an image. Strategies based on Fourier analysis and skillful use of successive radio-frequency pulses are used to go further and distinguish the signal from each point within the selected cut. With this type of strategy, it is possible to select and read the signal of each point, thus solving the inverse problem of the reconstruction of the object.

As said, a radio-frequency pulse at the Larmor frequency has the ability to excite the spins, which were initially, mostly, parallel to the field. Specifically, if the magnetization of the sample due to the field was initially parallel to B_0, the radio-frequency pulse will provoke a perpendicular magnetization to the field (3.3). After the magnetization has been rotated into the transverse plane, it will tend to grow back along the direction $\overrightarrow{z}$ of the static field $\overrightarrow{B_0}$. This is called *spin-lattice decay* or relaxation. This rate of regrowth can be characterized by a time constant T1 called the longitudinal relaxation time. The subsequent regrowth of the magnetic field is given by

$$M_z(t) = M_0(1 - e^{-t/T1}) \tag{3.5}$$

This return to the initial position emits an electromagnetic signal that can be detected externally, and which depends on the local chemical environment. By measuring this signal, the image can be formed, where the intensity at each point, obtained from the relaxation time of the spins in the longitudinal direction, is also informative of the tissue.

It can be deduced from Figure 3.2 that when the signal is acquired during the relaxation process, there is an instant T1 when each tissue presents a different signal level. If we acquire an image focusing on that moment, we will be able to maximize the contrast between the different tissues, since each one will be assigned a level of brightness depending on the signal it provides.

Another relaxation parameters affected differently by the tissue at each point can also be measured in order to visualize other properties of tissues.

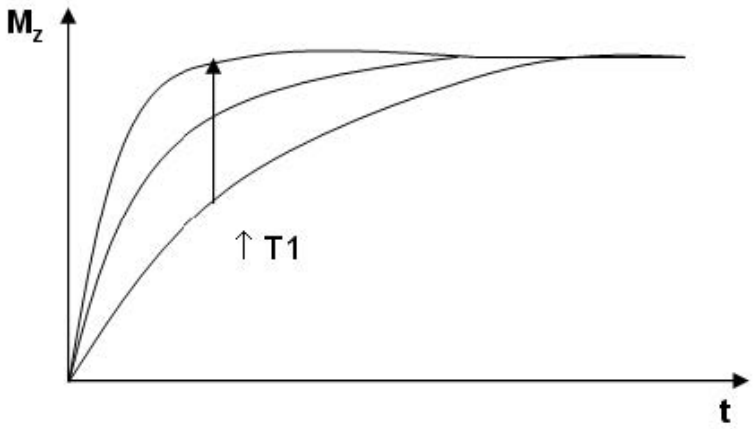


Figure 3.2: Longitudinal relaxation curves depending on the T1 value of the tissue.

One of them is the dephasing of clusters of spins, which represents a *spin-spin decay* of the transverse magnetization before data sampling can occur. Think about a situation where a $pi/2$ radio-frequency pulse is applied at an interval time of T_R. Any previous transverse magnetization has decayed away due to the spin-spin effect, and only the longitudinal magnetization (3.5) remains to be rotated into the transverse plane. If the signal data are instantaneous sampled at an *echo time* (TE) following the pulse, then

$$M_{\perp}(T_E) = M_0(1 - e^{-T_R/T1})e^{-T_E/T2} \tag{3.6}$$

The term $e^{-T_E/T2}$ represents the spin-spin decay factor characterized by the time constant T2, which is caused by a loss of coherence between different spins.

The spin density and the relaxation times T1 and T2 are three essential tissue parameters that allow generating three different image modalities that are going to be used in this research. Each of them has its characteristics and advantages for analyzing diseases.

3.2.2 Storage format

There are different formats for storing an MR image, but we will only focus on those that are commonly used. The standard format that all device manufacturers convert the magnetic resonance image acquisition is the Digital Imaging and Communication in Medicine (DICOM) format (Bidgood et al., 1997). It is the world-renowned standard for the exchange of medical tests, designed for handling, viewing, storage, printing, and transmission. It includes the definition of a file format and a network communication protocol. The communication protocol is an application protocol that uses TCP/IP

for communication between systems. DICOM files can be exchanged between two entities that have the ability to receive images and patient data in DICOM format.

DICOM enables the integration of scanners, servers, workstations, printers, and multi-vendor network hardware into an image storage and communication system. Different machines, servers, and workstations have a DICOM (conformance statements) declaration of conformity that establishes the DICOM classes they support. DICOM has been widely adopted by hospitals and is making a foray into small office applications of dentists and physicians.

However, this is not the format we are going to work with, as it is very complex and difficult to deal with. Besides, as mentioned, DICOM has been introduced mainly for clinical uses, and therefore few, and particular programs are available to work with this format. However, developers and biotechnology specialists who do not have access to them require a more familiar and useful format for research and development of applications and models. The most common are ANALYZE (Robb et al., 1989), NIfTI or HDR/IMG (Todd-Pokropek et al., 1992), and MiND (Patel et al., 2010) formats. Since most of the datasets were created using the Neuroimaging Informatics Technology Initiative (NIfTI) standard, this is the format we will focus on.

The NIfTI file format was intended a decade ago as a replacement to the then widespread, yet problematic, ANALYZE 7.5 file format[1]. The main problem with the above format was the lack of adequate information about the orientation in space. Although many different image programs used the format, the lack of adequate information about orientation forced some programs to include an attachment describing the orientation.

The new format was defined with the help of representatives of some of the most outstanding neuroimaging programs, who agreed on a format that included the new information. Perhaps the most visible consequence of the lack of orientation information was the then-reigning confusion between the left and right of brain images during the years in which the ANALYZE format predominated. The NIfTI format avoided all these issues. The data stored in NIfTI format also uses a pair of files as ANALYZE, *.hdr/.img*. However, working with a couple of files for each image as in the *.hdr/.img*, rather than just one, is not only an inconvenience, but it is also error-prone, as one could easily forget (or not know) that the data of interest is actually divided through more than one file. In order to address this topic, the NIfTI format also allows storage as a single file, with a *.nii* extension.

The header size is 348 bytes. Then, in Table 3.1, we show only some of the fields with a brief description, as the rest do not fall within the scope of the research.

[1]NIFTI: http://brainder.org/2012/09/23/the-nifti-file-format/

Table 3.1: Descrition of the NIfTI file header

Type	Name	Offset	Length	Description
int	sizeof_hdr	0B	4B	Length of the header
short	dim[8]	40B	16B	Array of dimensions
short	datatype	70B	2B	Type of the data
short	bitpix	72B	2B	Number of bits per voxel
float	pixdim[8]	76B	32B	Voxel dimensions
float	vox_offset	108B	4B	Offset within .nii file
char	xyzt_units	123B	1B	Units of pixdim[1..4]
float	cal_max	124B	4B	Maximum intensity
float	cal_min	128B	4B	Minimum intensity
float	slice_duration	132B	4B	Frame time
float	toffset	136B	4B	Frame duration

3.2.3 Limitations

There are several technical difficulties associated with MRI. Obtaining an ideal noiseless and high-resolution image is still a challenge. Also, there are human limitations that are also present in the acquisition and have to be considered.

The goal of a well-conducted MRI study is to obtain enough voxel signal relative to noise, also called SNR, and measured as the ratio of the voxel signal to the noise standard deviation, in order to observe tissues of interest. Generally, the noise derives from random fluctuations in the receive coil electronics and the sample. There are other sources of noise, such as digitization noise and pseudorandom ghosting due to moving spins that may affect the reconstructed image. It has been shown that MRI noise is governed by a specific distribution, the Rice distribution (Macovski, 1996). This type of noise is difficult to be modeled and removed from the images, so specific and complex works have been published for this task.

Another crucial aspect of MRI is its resolution. The signal dependence on the sample volume is crucial to very high-resolution imaging since the signal in each voxel is limited by its volume and the available magnetization for the given field. The inherent resolution in MR is a function of the way the signal and noise are sampled and filtered, and it is limited by the local magnetic field non-uniformities around the proton and the diffusion of the protons through the tissue. Besides, the acquisition time also limits the signal that can be collected, thus affecting noise and resolution. The position (and other characteristics) of the coils also play a role. In general, as the resolution is increased, the available signal decreases. The smaller the individual volume (voxel), the less signal it will produce. In order to obtain enough signals from the tiny volumes, higher fields need to be employed.

The voxel dimension uniformity is another feature related to the resolution to take into account. Traditionally, the size of a voxel is the same for the three dimensions (isotropic), although, for clinical applications, it is common to generate non-uniform (anisotropic) voxels. In those cases, one of the planes is generated with fine details (commonly seen in T2 images), while the remaining dimension has a large sampling size making indistinguishable the other two planes. However, isotropic voxels are preferred and not always can be achieved due to the technical limitations of the machine.

Another critical problem of MRI is the presence of an unexpected and even bothersome image inaccuracy or an artifact (Smith and Nayak, 2010). An artifact is defined as a false feature in the image created by some imperfect process in the data collection or the inverse Fourier transform that models the image reconstruction. For example, the discretization of infinite data may produce specific aliasing image errors. However, these artifacts may also be generated in the post-processing stage, for instance, when we want to remove the image noise of blur.

Regarding the human aspects, it is worth noting that the acquisition of a high-quality image is closely related to the time that the subject has to be immobilized inside the machine. Therefore, movements of the patient can produce distortions in the signal measurement, and the overall scanning time needs to be reduced. This leads to producing images of undesirable quality.

To sum up, there are essential aspects that are present in MRI, and many pieces of research are studying and can be grouped into two: noise and resolution. This research tries to contribute to providing new mechanisms for image enhancing.

Chapter 4

Learning techniques

It is easier to change the specification to
fit the program than vice versa.
Alan Perlis

Abstract: In this chapter, the different methodologies used to reach out to the goals of the research are described. First of all, the basic theory of function optimization is detailed, and then the well-known gradient descent method to reach the minimum of a function is described. Finally, the fundamentals of two powerful learning techniques are summarized since they are present in most of the works.

4.1 Function optimization background

The present research comprises many methodologies to enhance some of the existing methods for segmentation, point fitting, and image enhancing. All of them have one thing in common, the optimization of a function f to minimize or maximize the error defined by this function. Without loss of generality, one can assume that all the optimization problems are devoted to finding a minimum since it is possible to establish an equivalence between maximization and minimization problems by changing the sign of the optimization function, that is, $\min\{f\} = \max\{-f\}$.

Assuming the above, a formal definition about the concept of optimization problem is presented a follows:

Definition 4.1. (Optimization problem) An optimization problem is defined as a pair (A, f), where $A \subseteq \mathbb{R}^n$, $A \neq \emptyset$ is the solution space of the problem, and $f : A \to \mathbb{R}$ is the objective function. A solution of the optimization problem is given by an element $x^\star \in A$ such that:

$$f(x^\star) \leq f(x),\ \forall x \in A. \tag{4.1}$$

The function f is formally called an *objective function*. In machine learning, it receives the name of *cost function* since it is always necessary to continuously evaluate the quality of a data model where a minimum implies a set of possibly optimal parameters with an optimal error. A is called the *search space*, and it is usually a subset of the Euclidean space, often specified by a set of constraints.

The previous definition determines the finding of a value $x^\star$, which is called the global minimum solution of the optimization problem. Nevertheless, it is possible to stack into a neighborhood where the inequality (4.1) holds but the solution in not minimum in the search space A:

$$\exists \delta > 0 \text{ such that } \forall x \in A,\ \|x - x^\star\| \leq \delta \tag{4.2}$$

In this situation the value $x^\star$ is called a *local minimum*. Generally, there may be several local minima in a minimization problem unless the objective function is convex.

Remark 4.1. Given an open convex set A, a function $f : A \to \mathbb{R}$ is said to be convex if, and only if, $\forall\, x, y \in A$ and for all $t \in [0, 1]$, then $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$. Geometrically, this means that the segment determined by any two points of the graph of f is above the graph of the restriction of f to the segment $S_{x,y}$.

On the other hand, the simple one-objective optimization is not always the best method to achieve our goals since the problem may have more than one objective function to be optimized. One would desire to optimize two or more cost functions in order to find the optimal solution for more that one criteria simultaneously. This is called a *multi-objective optimization problem* and it can be formulated as

$$\begin{aligned} &\text{minimize } f_1(x), f_2(x), \ldots, f_k(x) \\ &\text{subject to } x \in A \subseteq \mathbb{R}^n \end{aligned} \tag{4.3}$$

where $f_i : A \to \mathbb{R}$, $i = 1, \ldots, k$, are each of the objective functions.

In general, (4.3) has many optimal solutions with different trade-offs. The set of trade-off solutions that improve upon one criterion at the expense of another is known as the *Pareto optimal set*. A variable $\tilde{x} \in A$ is called Pareto optimal if it is not dominated by any other design. Mathematically, $\nexists x \in A$ such as $f_i(x) \leq f_(\tilde{x})$, $\forall\, i = 1, \ldots, k$, and $\exists\, j$ such that $f_j(x) < f_j(\tilde{x})$. In Figure 4.1 is presented a schematic example of the Pareto set. The decision maker is in charge of making the choice among Pareto optimal solutions to determine the favorite solution.

4.2 The Gradient Descent method

In order to solve problems, iterative procedures are commonly used to converge to a solution in a finite number of steps. The Gradient Descent (GD)

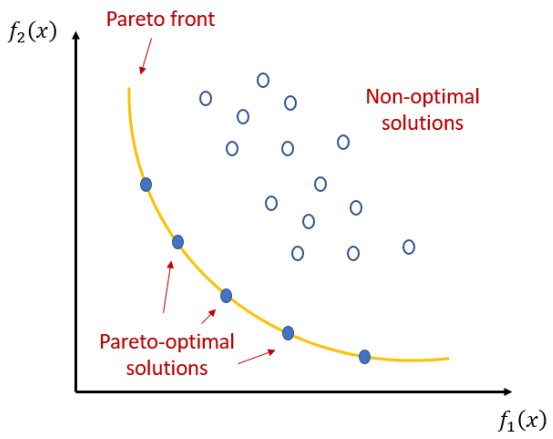


Figure 4.1: Illustration of the Pareto set of a two-function optimization problem.

method is a prevalent computational optimization technique for finding a minimum of a convex function.

Proposition 4.1. *Let A be a convex set of $\mathbb{R}^n$ and $f : A \to \mathbb{R}$ a differentiable and convex function. If $a \in A$ is a critical point of f, then a is an absolute minimum of f.*

To minimize a convex function, we need to find a stationary point. The GD approach starts at an arbitrary point, it moves along the gradient at that point towards the next point, and this procedure is repeated until converging to a stationary point. An illustrative example is a person who wants to get down a mountain. It tends to descent in the direction of the highest slope until it reaches the valley. The idea is the same.

Let $f : A \to \mathbb{R}$ be a convex function defined on a set $A \subseteq \mathbb{R}^n$. Starting at an arbitrary point $\mathbf{x}^{(0)}$, at every step $k \geq 0$ we move iteratively at the direction $\Delta\mathbf{x}^{(k)}$:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + s_k \Delta\mathbf{x}^{(k)} \tag{4.4}$$

where s_k is the step size.

In the gradient descent technique, we run for minimizing the function, so the negative gradient at the point is used, $\Delta\mathbf{x} = -\nabla f(\mathbf{x})$. Therefore, the iterative searching rule of gradient descent can be described as:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - s_k \nabla f(\mathbf{x}^{(k)}) \tag{4.5}$$

The choice of the step size value can be a fixed value or can be determined at each step to tuning up the minimization. In some cases finding s_k is computationally expensive, and different methods are used. One of the advantages of GD is its convergence. It has been demonstrated that in

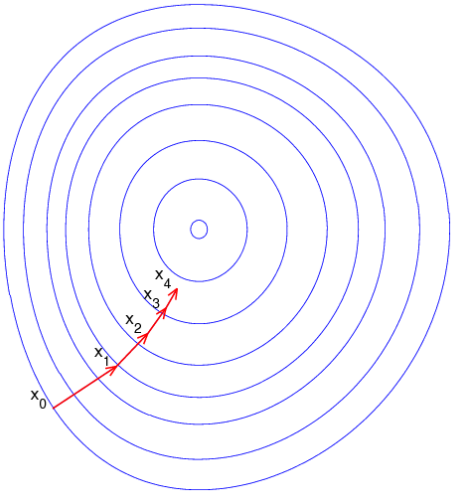


Figure 4.2: Illustration of the Gradient Descent method in two dimensions.

certain conditions (strongly convex functions), the gradient descent method always converges to the optimal solution.

On the other hand, GD can be slow to run on large datasets. This is because one iteration of the gradient descent algorithm requires a prediction for each instance in the training dataset. If we have many millions of instances, the algorithm not finish for a long time. In those situations when large amounts of data need to be processed, such as in deep learning (Section 4.4), a variation of gradient descent called Stochastic Gradient Descent (SGD) is the right choice.

Consider the following problem of minimizing an objective function that has the form:

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} F_i(w) \tag{4.6}$$

where w is the parameter two be estimated and the functions F_i are associated to the i sample of the dataset.

In this variation, the update is carried out for each training instance rather than at the end of the batch of instances. The adaptation rule is still the same, and the cost is not added among all training patterns, but instead calculated for one training pattern.

$$w^{(k+1)} = w^{(k)} - s_k \frac{1}{n}\sum_{i=1}^{n} \nabla F_i(w^{(k)}) \tag{4.7}$$

The iterative algorithm requires that the order of the training dataset is randomized. This is to mix up the order that updates are made to the estimated parameter, so noisy jumps are flattened, and stagnation is avoided.

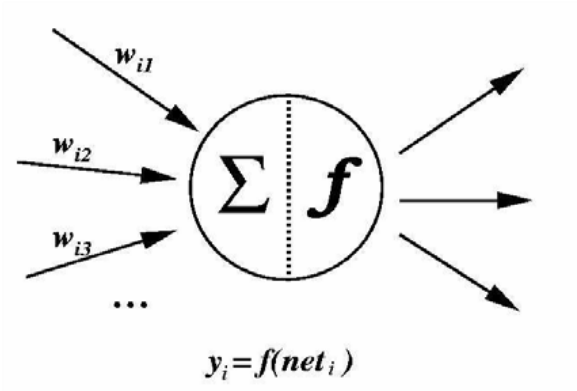


Figure 4.3: Illustration of an artificial neuron.[1]

Usually, a small number of passes through the dataset are needed to reach the optimal point.

These techniques is present in this research for both point fitting and deep learning (Parts III and IV).

4.3 Artificial neural networks

Artificial neural networks were originally a computational model of biological nervous systems, made up of a set of units called artificial neurons or nodes connected together. The first neural network model was proposed in 1943 by McCulloch and Pitts in terms of a computational model of nerve activity (Pitts and McCulloch, 1947). This model was a binary model, where each neuron had a prefixed step or threshold and served as the basis for subsequent models. The applied neural networks are, in general, inspired by biological neural networks, although they have other functionalities and connection structures that are different from those seen from a biological perspective.

The basic computing element is usually called a node or unit (see Figure 4.3). It receives input from other units or an external data source. Each input has an associated *weight* w_i, which is modified in the so-called learning process, combine the input with their internal state and an optional threshold using an activation function. Then, each unit applies a *transfer function* f to the combination of the inputs weighted by the weights to produce an output.

Artificial neural networks are usually composed of too many neurons. The *weight* of each neuron represents its relative importance. A given neuron can have multiple inputs and one or more outputs, which are called *connections*. All the neurons that are organized at the same level receive the

[1] Extracted from https://mathengineering.blogspot.com/2017/01/redes-neuronales-artificiales-ann.html

name of *layer*. Typically, there are multiple layers divided into three blocks: the input layer, the hidden layers, and the output layer. This is specially relevant in deep networks (see Section 4.4). Between layers, the connection can be complete or partial; even some architectures allow connections between the same or previous layers in a recurrent way.

Neural networks are commonly classified in terms of their corresponding training algorithms or methods: unsupervised networks and supervised training networks. In supervised training neural networks, the data for training consists of several pairs of input and output training patterns. For example, a widely used criterion in so-called supervised training is to minimize the root mean square error between the output value and the expected actual value. For unsupervised training models, the training data set consists only of the input patterns. The network learns to adapt to experiences gathered from previous training patterns.

There are two phases in modeling with neural networks:

- *Training phase.* A training dataset is used to determine the weights that define the neural network model. They are calculated iteratively, and typically the optimal weights are obtained by optimizing some energy function.

- *Test phase.* In the previous phase, the model may be specialized in the features present in the training patterns, losing its ability to generalize its learning to new cases (*overfitting*). In order to avoid this problem, it is advisable to use the second group of data other than training data, the validation group, which allows controlling the learning process.

Three very important factors in the training phase are the *learning rate*, the *backpropagation*, and the *cost function*. The first one defines the size of the correction that the model takes to adjust for errors in each iteration. Backpropagation is the technique to adjust the weights to compensate for each error found during the learning step. Finally, the cost function refers to an ad hoc function whose value can only be approximated.

An adequate choice of the cost function enhances the behavior of the neural network, and this is one of the goals of this research. The essential tools about how the cost functions will be defined are detailed in Chapter 5. Classical unsupervised neural network models such as competitive neural network or neural gas network are going to be used in this research for segmentation tasks in Chapters 6 and 7.

4.4 Deep learning

A deep learning architecture is a multi-layered neural network with numerous hidden layers. Many of these layers calculate non-linear functions, which, in

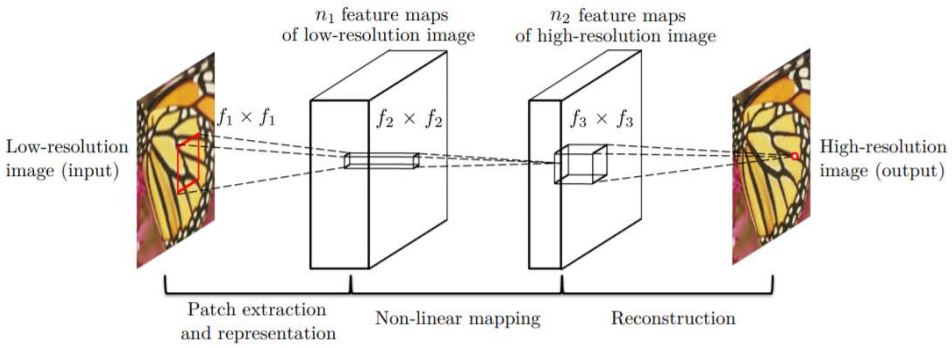


Figure 4.4: Structure of Super-Resolution Convolutional Neural Network (SRCNN) deep neural network.[2]

the case of image analysis, make the system sensitive to small details and insensitive to large variations such as lighting, background, or surroundings of objects. This type of network is not only applicable to image analysis but has also been used successfully in other fields such as automatic speech recognition.

The importance of Deep Learning (DL) is that it is widely used nowadays because it is obtaining significantly better results than with traditional shallow networks, and it is bringing about a revolution in the way in which most problems related to image analysis were solved. This has been possible due to the higher computational capacity of today's computers and, above all, to the great innovation in graphic cards and their Graphic Processing Unit (GPU) that they have incorporated, and more specific hardware developments (Ortega-Zamorano et al., 2017). Nevertheless, there is a significant dependency on the availability of enough data for training. The deeper the network, the more data is needed to train adequately the neurons of all the layers, which can reach many millions.

The most popular architecture in image analysis is the one that corresponds to the Convolutional Neural Network (CNN), which uses several convolutional layers combined with other types of layers. The general idea is that convolutional layers are capable of extracting sophisticated features from the image while reducing the dimensionality of the processed data. CNNs have demonstrated excellent performance in image and video processing. These methods are under constant development (Rafiei and Adeli, 2017; Hua et al., 2019; Koziarski and Cyganek, 2017), and they have been successfully applied in detection and recognition of objects, classification of images, or within recommender systems (Molina-Cabello et al., 2018; Wang and Bai, 2018). Hundreds of articles based on the development of CNNs have been published in several areas (Liu et al., 2017), including medical image analysis

[2]Extracted from Dong et al. (2016)

(Prince et al., 2019; Zhao et al., 2019; Shi et al., 2019; Li et al., 2019), where the use of CNNs is progressively expanding.

Within CNNs, there are several types of architectures, implementing different numbers and types of layers, kinds of output, etcetera. Many high-performance CNNs can be found in the literature such as AlexNet (Krizhevsky et al., 2017), VGG (Simonyan and Zisserman, 2015), GoogleNet (Szegedy et al., 2015), ResNet (He et al., 2016), etc. Some, like (Graepel, 2016), have been shown to exceed the precision of the human visual system in certain object recognition tasks. In this research, deep networks are going to be used for medical applications, although the use of these models is still low because of the need for large datasets to achieve good performances. This limitation is going to be overcome by different techniques based on the ℓ_p-norm.

Focusing on super-resolution, one of the first deep networks made for this task was SRCNN (Dong et al., 2016). Its architecture is displayed in Figure 4.4. It stands out for its simplicity since it only contains two hidden layers, but the patch extraction methodology and non-linear mapping makes this net very accurate. The mapping operation non-linearly maps each high-dimensional vector onto another high-dimensional vector and allows it to comprise the set of fundamental features. Regarding its cost function, the usual Mean Squared Error is used.

The central part (Part IV) of our research tries to exploit the benefits of the 3D version of this deep network, and the new models have been built around it.

Chapter 5

Normed spaces and Measures

I know that I know nothing
Socrates

Abstract: The main theoretical tools of this research are presented in this chapter. Normed spaces provide a basis for a rich theory, but they are also general enough to include many concrete models of practical importance. The definition of the cost functions is essential for determining the goal of the minimization procedure. The Euclidean norm is the most-used method to measure errors, although it has its advantages and disadvantages. Robustness is commonly reached out with the absolute norm, and the use of another ℓ_p-norms has to be explored.

5.1 Normed space and Banach space

Let give some definitions in order to understand the basis of the error measures used in this research.

Definition 5.1. (Normed space) A *normed space* X is a vector space with a norm defined on it. A norm on X is a real-valued function $\|\cdot\| : X \to \mathbb{R}$ with the following properties:

1. $\|x\| \geq 0$
2. $\|x\| = 0 \iff x = 0$
3. $\|\alpha x\| = |\alpha| \, \|x\|$
4. $\|x + y\| \leq \|x\| + \|y\|$

being $x, y \in X$ and $\alpha \in \mathbb{R}$.

A norm on a normed space induces a distance d, which is given by $d(x, y) = \|x - y\|$. The normed space can be noted as $(X, \|\cdot\|)$. More generally, if in a set X we define any distance d, then is called a *metric space* and noted as (X, d).

Definition 5.2. (Banach space) A Banach space is a complete normed space.

The notion of completeness is reached through the convergence of Cauchy sequences:

Definition 5.3. (Cauchy sequence) A sequence (x_n) in a metric space (X, d) is said to be a *Cauchy sequence* if $\forall \varepsilon > 0$, $\exists N$ such that $d(x_n, x_m) < \varepsilon$, $\forall n, m > N$.

Definition 5.4. (Completeness) The space X is said to be *complete* is every Cauchy sequence in X converges in X (the limit belongs to X).

The Banach spaces are metric spaces, and they have many interesting properties. First of all, the L^2, L^1, and L^p spaces that are going to be defined can be completed to be Banach spaces on all continuous real-values functions (Bramwell and Kreyszig, 1979). The Hahn-Banach theorem guarantees that a normed space is supplied with bounded linear functionals, so dual spaces can be theorized, which is an essential part of the general theory of normed spaces. For example, the dual of a Banach space is also a Banach space. Thus, as $L^p([0, 1])$ is isometrically isomorphic to $L^q([0, 1])$ when $1 \leq p < +\infty$ and $1/p + 1/q = 1$, for all p, we can define a Banach space. Banach spaces also give characterizations for weak convergence of sequences through the uniform boundedness theorem. This is of interest for its application to the L^1 space. Both the open mapping theorem and the closed graph theorem are another essential results of the Banach theory.

The following sections introduce the concepts of norms. We will see that the standard ℓ_2-norm is not robust, and it must first be considered as the base option from which ℓ_1-norm and ℓ_p-norm with $p < 2$ are proposed as more robust techniques than ℓ_2-norm. The mean is associated with ℓ_2-norm, and the median with ℓ_1-norm.

5.2 L^2 space

Definition 5.5. ($L^2[a, b]$ space) The vector space $L^2[a, b]$ of all continuous real-values functions on $[a, b]$ is a normed space with norm defined by

$$\|f\|_2 = \left(\int_a^b f(x)^2 dx \right)^{1/2} \tag{5.1}$$

More specifically, our works will manage points and pixel (or voxel) information, which are in essential real vector on $\mathbb{R}^n$. Thus, it would be adequate to define a space on the metric space $\mathbb{R}^n$. Next, we define this space and its norm, as well as the mean value, which has an interesting property associated with the Euclidean norm.

5.2.1 ℓ_2 space and ℓ_2-norm

The ℓ_2 *space* or *Euclidean space* on $\mathbb{R}^n$ is a vector space defined with the usual algebraic operations (sum and product by a scalar) and with a norm defined by

$$\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2} \tag{5.2}$$

which is commonly named as the ℓ_2-norm or Euclidean norm.

Therefore, the Euclidean distance is defined as

$$d_2(x, y) = \|x - y\|_2 = \left(\sum_{i=1}^{n} (x_i - y_i)^2\right)^{1/2} \tag{5.3}$$

The ℓ_2-norm is the most commonly used measure because it is easy to calculate and is easily understood. Also, it lends itself to algebraic operations.

5.2.2 The mean

As an application to the general function optimization problem, we find out that the mean minimizes the squared error function (Farebrother, 2013).

Let $y_1, y_2, \ldots, y_n$ represent a set of n observations on a single variable Y defined on $\mathbb{R}^n$. We want to identify a point of best fit to these n points in such a way that the following cost function is minimized:

$$E(y_0) = \sum_{i=1}^{n} \|y_i - y_0\|_2^2 = \sum_{i=1}^{n} (y_i - y_0)^2 \tag{5.4}$$

The following proposition can be proved:

Proposition 5.1. *The mean is the point that minimizes the sum of the squared distances (ℓ_2-norm) the from each sample.*

Proof. The squared error function is convex and differentiable. This conditions ensure that the minimum is unique, and of course, that the gradient exists. Thus,

$$\nabla E(y_0) = 2 \sum_{i=1}^{n} (y_i - y_0)$$

From the necessary conditions of optimality follows that the gradient vanishes at the unique minimizer:

$$\nabla E(y_0) = 2\sum_{i=1}^{n}(y_i - y_0) = 0$$

Therefore, solving this equation:

$$2\sum_{i=1}^{n}(y_i - y_0) = 0 \Rightarrow \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} y_0 = 0 \Rightarrow \sum_{i=1}^{n} y_i = n \cdot y_0 \Rightarrow y_0 = \frac{1}{n}\sum_{i=1}^{n} y_i$$

□

The arithmetic mean has advantages and disadvantages. The advantages are:

- It is the most widely used measure of central tendency.
- It uses all the available information in its calculation.
- It is expressed in the same units as the variable under study.
- It is a unique value.
- It is sensitive to any changes in the data (it can be used as a detector of variations in the data).
- It is often used in subsequent statistical calculations.
- It is useful for performing statistical procedures such as comparing means of various data sets.
- On the frequency graph, it represents the center of gravity.

On the other hand, the following problems are associated with the mean:

- It is sensitive to extreme values.
- It is not recommended to use it in very asymmetric distributions.
- If discrete variables are used, then the arithmetic mean may not belong to the set of values of that variable.
- It cannot be calculated for qualitative data.
- The computed value is highly affected by extreme values or outliers.

5.3 L^1 space

Analogously, instead of using the squares of the vector components, we might consider the use of their absolute values. This formulation does not enhance the weight of each component, and its definition is quite simple.

Definition 5.6. **($L^1[a,b]$ space)** The vector space $L^1[a,b]$ of all continuous real-values functions on $[a,b]$ is a normed space with norm defined by

$$\|f\|_1 = \int_a^b |f(x)|\,dx \tag{5.5}$$

As before, we are in the situation of defining the vector space and its norm.

5.3.1 ℓ_1 space and ℓ_1-norm

The ℓ_1 *space* on $\mathbb{R}^n$ is a vector space defined with the usual algebraic operations and with norm defined by

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \tag{5.6}$$

which is commonly named as the ℓ_1-norm.

Therefore, the distance defined in this space is formulated by

$$d_1(x, y) = \|x - y\|_1 = \sum_{i=1}^{n} |x_i - y_i| \tag{5.7}$$

The main advantage of the ℓ_1-norm compared to the ℓ_2-norm is that it is more robust when the data values are dispersed. However, it has the penalty that it is not perfectly differentiable. Besides, the ℓ_2-norm is more used because it has the valuable property that it is invariant under rotations, which is useful for many applications (like Principal Component Analysis).

5.3.2 The median

A stretch relationship with the ℓ_1-norm is the median, which minimizes the sum of absolute deviations.

In this case, we want to find a point of best fit to a set of points in such a way that the following cost function is minimized:

$$E(y_0) = \sum_{i=1}^{n} \|y_i - y_0\|_1 = \sum_{i=1}^{n} |y_i - y_0| \tag{5.8}$$

The following proposition can be proved:

Proposition 5.2. *The median is the point that minimizes the sum of the absolute distances (ℓ_1-norm) the from each sample.*

Proof. Assume, without loss of generality, that the set of points $y_1, y_2, \ldots, y_n$ is ordered:

$$y_1 \leq y_2 \leq \cdots \leq y_n$$

Observe that (5.8) can be rewritten as follows:

$$\sum_{i=1}^{n} |y_i - y_0| = \sum_{i=2}^{n-1} |y_i - y_0| + (y_n - y_1),\ y_0 \in [y_1, y_n] \tag{5.9}$$

We distinguish between two cases:

- n is odd: applying the above identity (5.9) repeatedly

$$E(y_0) = \sum_{i=1}^{n} |y_i - y_0| = \left| y_{\frac{n+1}{2}} - y_0 \right| + (y_n - y_1) + (y_{n-1} - y_2) + \cdots + (y_{\frac{n+3}{2}} - y_{\frac{n-1}{2}})$$

That is,

$$E(y_0) = \left| y_{\frac{n+1}{2}} - y_0 \right| + C$$

where C is a constant.

Now, taking the derivative:

$$\nabla E(y_0) = \operatorname{sign}\left(y_{\frac{n+1}{2}} - y_0 \right)$$

And this vanishes for $y_0 = y_{\frac{n+1}{2}}$, which is the median of a set of an odd number of points.

- n is even: by using (5.9)

$$E(y_0) = \sum_{i=1}^{n} |y_i - y_0| = \left| y_{\frac{n}{2}} - y_0 \right| + \left| y_{\frac{n+2}{2}} - y_0 \right| + C$$

Differentiating and setting $\nabla E(y_0)$ to zero we get

$$\frac{\left| y_{\frac{n}{2}} - y_0 \right|}{y_{\frac{n}{2}} - y_0} + \frac{\left| y_{\frac{n+2}{2}} - y_0 \right|}{y_{\frac{n+2}{2}} - y_0} = 0$$

Observe now that the median $y_0 = \frac{y_{\frac{n}{2}} + y_{\frac{n+2}{2}}}{2}$ satisfies the equation since it is halfway between these two points and

$$y_{\frac{n}{2}} - y_0 = -(y_{\frac{n+2}{2}} - y_0)$$

Therefore, the median is the minimum.

□

Unlike the mean, the advantages of the median are:

- It is a robust position measurement. Extreme values do not influence it since only the central values matter.
- If there are censored data in the sample, it is not possible to calculate the mean; however, the median may eventually be calculated.
- Easy to understand.

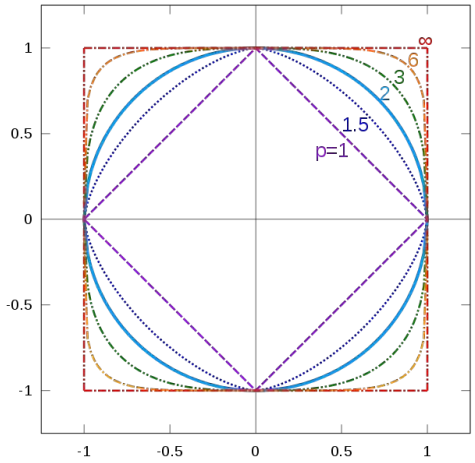


Figure 5.1: Illustration of some ℓ_p-norms in the interval $[-1, 1]$.

- It is the most representative measure of central tendency in the case of variables that only admit the ordinal scale, so not only numeric data are admitted.
- Easy to compute if the number of observations is not very big.
- It is useful for performing statistical procedures such as comparing medians of various data sets.

On the opposite side, there are some disadvantages:

- It is not always a unique value.
- It does not use all the available information in its calculation.
- It is insensitive to the distance of the observations to the center since it only depends on the order of the data.
- The data need to be sorted before determining it.
- It is difficult to calculate if the number of observations is huge.

5.4 L^p space

Finally, and more generally, the previous Banach spaces can be extended for any fixed real number $p \geq 1$ as follows:

Definition 5.7. ($L^p[a, b]$ space) The vector space $L^p[a, b]$ of all continuous real-values functions on $[a, b]$ is a normed space with norm defined by

$$\|f\|_p = \left(\int_a^b |f(x)|^p \, dx \right)^{1/p} \tag{5.10}$$

The condition $p \geq 1$ is necessary to ensure that the definition of (5.10) defines a norm.

5.4.1 ℓ_p space and ℓ_p-norm

The ℓ_p *space* on $\mathbb{R}^n$ is a vector space defined with the usual algebraic operations and with the ℓ_p-norm defined by

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{5.11}$$

This norm induces the distance

$$d_p(x, y) = \|x - y\|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{5.12}$$

As p tends towards $+\infty$, the ℓ_p-norm tends towards the max function, which means that we are essentially penalizing the largest argument of the norm, i.e., the largest difference between components if we measure distances. This fact can be seen in Figure 5.1. Another important behavior is that for larger p values, undesirable vector components can hide under the largest one, allowing the generation of less-sparse models since the computed distances are not as much affected by those undesirable values as with lower ℓ_p-norms like with $p = 1$, which gives an equal penalty to all components.

Part II

Segmentation

Chapter 6

Panoramic background modeling for PTZ cameras with competitive learning neural networks

The best way to predict the future is to create it.

Peter Drucker

Abstract: The construction of a model of the background of a scene still remains as a challenging task in video surveillance systems, in particular for moving cameras. This work presents a novel approach for constructing a panoramic background model based on competitive learning neural networks and a subsequent piecewise linear interpolation by Delaunay triangulation. The approach can handle arbitrary camera directions and zooms for a Pan-Tilt-Zoom (PTZ) camera-based surveillance system. After testing the proposed approach on several indoor sequences, the results demonstrate that the proposed method is effective and suitable to use for real-time video surveillance applications.

6.1 Introduction

With their high mobility and zoom capability, Pan-Tilt-Zoom cameras have become increasingly popular in monitoring public areas (Chen et al., 2009;

Ding et al., 2012; Konda et al., 2016). Omnidirectional cameras are promising candidates for monitoring multiple latent activities in the area of interest (Boult et al., 2004). However, these kind of cameras have nonuniform resolution and are unable to provide close observations of particular targets. In these cases, where PTZ cameras are needed, the combination of these two types of cameras (omnidirectional and PTZ) is proposed in order to facilitate a continuous monitoring of the whole surveillance area and detailed observations of specific targets simultaneously (Chen et al., 2008). Nevertheless, this dual-camera system may be still an expensive and complex solution in some scenarios. For this reason, in this research we are focusing on an active sensing approach to multiple object detection and tracking using a single PTZ camera.

The problem of moving objects detection for PTZ cameras is addressed in this research, and we propose a method based on building a panoramic background model using a competitive neural network. Apart of the traditional and frequently cited seminal papers related to competitive learning (Hofmann and Buhmann, 1998; Uchiyama and Arbib, 1994; Gersho, 1992), recent successful applications in the computer vision field can be found in the literature (Peng Chen et al., 2016; Ozan et al., 2016; Valente and Abrão, 2016; Xie et al., 2016). In our approach, a competitive neural network is used to build a panoramic background model for object detection. Due to the huge input information, a large number of neurons has been used and the neuron prototypes have been organized in a quad-tree in order to be quickly evaluated.

The rest of the chapter is organized as follows. In section 6.2, a more detailed description of the proposed neural model is presented. In section 6.3, we present the results achieved with the implementation of the proposed approach. Finally, section 6.4 includes some concluding remarks.

6.2 The models

In this section a competitive learning based system to learn the background of a panoramic scene from the input of a PTZ camera is proposed. First the data acquisition procedure to transform the input video frames into input samples for the competitive learning network is considered (Subsection 6.2.1). Then the competitive learning model is described (Subsection 6.2.2). Finally, an interpolation procedure is designed to estimate the background from the final state of the competitive learning network, which is based on a Delaunay triangulation (Subsection 6.2.3).

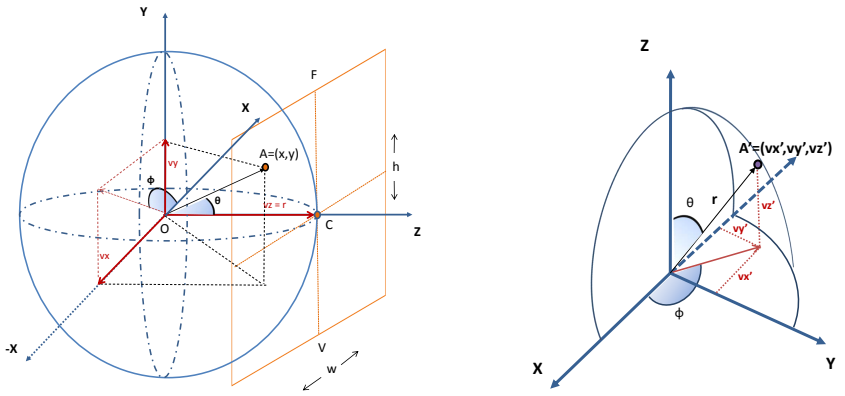


Figure 6.1: Schemes of the PTZ camera model. a) Left: pinhole camera model with the original orientation. b) Right: reoriented scheme to obtain spherical coordinates.

6.2.1 Data acquisition

The data from which we start are the acquired video frames of the PTZ camera. These are video frames with fixed width and height, and every pixel has two frame coordinates (x, y). To transform these frame coordinates to another coordinate system in the panoramic image, the first step is to obtain the spherical coordinates associated to the frame coordinates and then carry out a scalar transformation to the dimensions of the panoramic image.

In order to find out required polar coordinates (θ, ϕ) of an arbitrary point A belonging to projection plane (x, y) a pinhole camera model is used, consisting of a sphere centered on the coordinate origin and a projection plane (see Figure 6.1a), whose bounds are the width and height of the frame window in pixels: w, h. In this work the virtual PTZ library (Chen et al., 2015) is employed, so we must follow the coordinate system criteria used by that library. We start with the case of camera orientation $(pan, tilt) = (0, 0)$, which corresponds to see the positive Z axis from the origin. The general case, when $pan \neq 0$, $tilt \neq 0$ is obtained from the particular case by means of a coordinate system transformation.

The coordinates of vector OA in coordinate system O can be found as projections on the coordinate axes:

$$(vx, vy, vz) = (x - w/2, h/2 - y, r) \tag{6.1}$$

where $vz = r$, since points A and C are located on the same plane. vx and vy is calculated taking in account that A on the projection plane is based relatively of the left top corner of the frame. Radius r is calculated using the vertical field of view (FOV), which is known. We only need to notice

that $\widehat{FOC} = \widehat{FOV}/2$, and FOC is a right triangle, so

$$\tan(\widehat{FOV}/2) = \frac{FC}{r} \Leftrightarrow r = \frac{h/2}{\tan(\widehat{FOV}/2)} \tag{6.2}$$

This way the camera coordinates are computed. In order to extend the particular case to the generic case and obtain the world coordinates we need to recalculate vector OC from the O coordinate system into the rotated coordinate system O'. The coordinates are found by means of multiplication by the inverse of a double rotation matrix. This matrix can be found by multiplication of two rotation matrices: around X and Z axes by $(pan, tilt)$:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(tilt) & -\sin(tilt) \\ 0 & \sin(tilt) & \cos(tilt) \end{pmatrix} \cdot \cdot \begin{pmatrix} \cos(pan) & -\sin(pan) & 0 \\ \sin(pan) & \cos(pan) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.3}$$

$$A' = (vx', vy', vz') = R^{-1} \cdot (vx, vy, vz) \tag{6.4}$$

Now, when the new coordinates A' are computed, we only need the spherical coordinates transformation to obtain θ and ϕ:

$$\begin{cases} vx' = R\cos(\phi)\sin(\theta) \\ vy' = R\sin(\phi)\sin(\theta) \\ vz' = R\cos(\theta) \end{cases} \tag{6.5}$$

By rearranging the formulas:

$$\begin{cases} \phi = \arctan(\frac{vy'}{vx'}) \\ \theta = \arccos(\frac{vz'}{R}) = \arccos(\frac{vz'}{\sqrt{vx'^2+vy'^2+vz'^2}}) \end{cases} \tag{6.6}$$

The obtained samples are real valued vectors, whose two first components are the spherical coordinates ϕ and θ of a pixel, while the third last ones are the observed RGB color values at that pixel:

$$\mathbf{x} = (\theta, \phi, r, g, b) = (x_1, x_2, x_3, x_4, x_5) \tag{6.7}$$

so that $\mathbf{x} \in \mathbb{R}^5$. At each time instant t, a new episode $\mathcal{S}_t \subset \mathbb{R}^5$ is acquired which contains one sample for each pixel of the incoming video frame at time t. Therefore the cardinal of $\mathcal{S}_t$ is the number of pixels in the video frame.

6.2.2 Competitive learning network

Next a competitive learning model for panoramic scenes captured by PTZ cameras is developed. In order to learn the details of the scene, a large number of neurons N is employed, which is lower but in the same order of magnitude as the number of pixels of the full panorama. Following the strategy in (Luque-Baena et al., 2015), the input vectors $\mathbf{x} \in \mathbb{R}^5$ are divided into two sections. The first section contains the positional information in the video frame, while the second section contains the color features. In our case, the first section comprises the two spherical coordinates ϕ and θ, while the second section contains the RGB color values. All the components of the input vectors are used to update the neuron prototypes $\mathbf{w}_i \in \mathbb{R}^5$, but only the two first ones participate in the competition. This way, only the positional information is employed to determine which neuron is the winner. Therefore, each neuron represents the average color in a receptive field which is one region of a Voronoi tesellation of the full panorama. That is, the neurons specialize on small pixel neighborhoods of the panorama. Since the positional components can have fractional values, $(x_1, x_2) \in \mathbb{R}^2$, the network can learn fractional positional information without having to round to integer pixel positions in the panorama, which avoids losing valuable information.

Each neuron contains a prototype $\mathbf{w}_i \in \mathbb{R}^5$ and also a Boolean flag $b_i \in \{true, false\}$. The flag is required to control the initialization of the neuron. Since the PTZ camera does not cover the entire panorama at a time, it is not possible to initialize all the neurons at the same time. They can only be initialized while their positional components fall into the current field of view. To this end, the Boolean flags are initialized to false. The first time that a neuron wins, its prototype is set to the input sample, and its flag is set to true. From that point on, the neuron will be updated according to the competitive learning rule.

The proposed learning algorithm is as follows:

1. Draw a training sample $\mathbf{x}$ at random from from the current episode $\mathcal{S}_t$, which has not been considered before, i.e. a random sampling without replacement is done.

2. Find the nearest neuron q in terms of Euclidean distance according to the first two vector components:

$$q = \arg \min_{i \in \{1,\dots,N\}} \|(w_{i,1}, w_{i,2}) - (x_1, x_2)\| \tag{6.8}$$

3. If the flag b_q is true, then go to step 4. Otherwise, set the prototype of the winning neuron q to the training sample and set its flag to true:

$$\mathbf{w}_q = \mathbf{x} \tag{6.9}$$

$$b_q = true \tag{6.10}$$

Then go to step 5.

4. Update the winning neuron prototype according to the standard competitive learning rule:

$$\Delta \mathbf{w}_q = \eta(t)\,(\mathbf{x} - \mathbf{w}_q) \tag{6.11}$$

where $\eta(t)$ is a decaying learning rate that varies depending on the time step t.

5. If all the samples of the current episode $\mathcal{S}_t$ have already been processed, then go to step 6. Otherwise, go to step 1.

6. If the last time instant t has been reached, then stop. Otherwise, increment the time instant counter t, load the next episode and go to step 1.

There are two phases in the learning process: first the ordering phase where η experiences a linear decay (initial learning rate η_I); and then the convergence phase where η remains constant at a small value (η_C). This is because the ordering phase is required for the warm-up of the algorithm only, and after that the system runs for an indefinitely long time. The change of phase is monitored by a step parameter n.

$$\eta(t) = \begin{cases} \eta_I(1 - t/n), \text{ if } t < n \\ \eta_C, \text{ otherwise} \end{cases} \tag{6.12}$$

A theoretical analysis of the above algorithm can be carried out. Since the competition is done on the first two components, the algorithm seeks a local minimum of an energy function $\mathcal{E}$ which only takes into account these components:

$$\mathcal{E} = \sum_{i \in \{1,\ldots,N\}} \sum_{\mathbf{x} \in F_i} \|(w_{i,1}, w_{i,2}) - (x_1, x_2)\|^2 \tag{6.13}$$

where F_i is the receptive field of the i-th neuron:

$$F_q = \left\{ \mathbf{x} \mid q = \arg \min_{i \in \{1,\ldots,N\}} \|(w_{i,1}, w_{i,2}) - (x_1, x_2)\| \right\} \tag{6.14}$$

On the other hand, the prototype update is carried out on the three last components too, so those components approximate the average color of the receptive field:

$$(w_{i,3}, w_{i,4}, w_{i,5}) \approx E\left[(x_3, x_4, x_5) \mid F_i\right] \tag{6.15}$$

As a last remark, it must be pointed out that the large number of neurons N to be used for this application requires a considerable optimization of the competition equation (6.8). This is accomplished by inserting all the first sections $(w_{i,1}, w_{i,2})$ of the neuron prototypes into a quad-tree (Frisken and Perry, 2005). This way (6.8) is evaluated very quickly, even for values of N in the millions.

6.2.3 Delaunay triangulation based interpolation

Since the positions of the neuron prototypes are given by real numbers, there is no direct way to obtain the estimated color for the integer valued pixel coordinates of the panorama. In order to overcome this difficulty, we propose to build the Delaunay triangulation (Lee and Schachter, 1980) of the set formed by the first sections $(w_{i,1}, w_{i,2})$ of each neuron prototype. This way, a triangulation of the panorama is obtained. Then, for each integer valued pair of pixel coordinates $(y_1, y_2) \in \mathbb{N}^2$, the triangle which it belongs to is computed, along with its barycentric coordinates with respect to that triangle:

$$\mathbf{y} = \lambda_i \left(w_{i,1}, w_{i,2}\right) + \lambda_j \left(w_{j,1}, w_{j,2}\right) + \lambda_k \left(w_{k,1}, w_{k,2}\right) \tag{6.16}$$

$$i, j, k \in \{1, ..., N\} \tag{6.17}$$

$$\lambda_i, \lambda_j, \lambda_k \geq 0 \tag{6.18}$$

$$\lambda_i + \lambda_j + \lambda_k = 1 \tag{6.19}$$

Then the estimated color $(y_3, y_4, y_5) \in \mathbb{R}^3$ is obtained by linear interpolation with weights equal to the barycentric coordinates:

$$(y_3, y_4, y_5) = \lambda_i \left(w_{i,3}, w_{i,4}, w_{i,5}\right) +$$

$$\lambda_j \left(w_{j,3}, w_{j,4}, w_{j,5}\right) + \lambda_k \left(w_{k,3}, w_{k,4}, w_{k,5}\right) \tag{6.20}$$

Therefore a continuous, piecewise linear function is employed to estimate the color over the panorama.

6.3 Experimental results

In this section we report the computational experiments we have carried out and their results. The software and hardware that have been used are specified in Subsection 6.3.1. Then, the tested video sequences are described

in 6.3.2. The descriptions of the used parameters are in Subsection 6.3.3 and the descriptions of the competitors in 6.3.4. Finally, the obtained results from the experiments are reported in Subsection 6.3.5.

6.3.1 Methods

The camera control module is based on the *virtualptz library* (Chen et al., 2015). It simulates a PTZ camera from a panoramic video sequence, and it is accessible from its website [1]. The implementation is written in C++ and it uses the OpenCV and OpenGL libraries. This virtual camera has limitations in its vertical movement, going from 0 (up) to 180 (down) degrees.

To generate the input data for the competitive neural net and for the competitors, a exhaustive scanning of the scene has been carried out. To simulate the real behavior of a PTZ camera in a practical setting, some limitations in the scan have been imposed. Starting at the top (vertical 0 degrees), we turn 360 degrees to the right with a step of 10 degrees, and when we come back to the initial point, we go down 10 degrees. This process is repeated until we arrive to the bottom and we start to go up again. Furthermore, a random zoom (vertical field of view) is applied to each frame, but again, we try to simulate a real situation. To this end, we generate a random number in an interval $[FOV, FOV + 5]$ and move this interval by steps of 5 degrees between a minimum and maximum value for the field of view, $[70, 140]$ degrees, in order to avoid strange images with irregularities.

A total of 3000 camera frames have been saved in binary files to read them synchronously by all the methods, which have been implemented in MATLAB R2015b. The reported experiments have been carried out on a 64-bit Personal Computer with an eight-core Intel i7 3.60GHz CPU, 32 GB RAM and standard hardware. The implementation of our approach does not use any GPU resources.

Panoramic Ground Truth (GT) has been calculated doing the median of the raw panoramic video frames. For each method, we compared the panoramic image obtained with the ground truth. Three quality measures were used to evaluate the proposed approach: the first was the Mean Squared Error (MSE) metric (lower is better), which is commonly used in image processing; the second was the Structural Similarity Index Measure (higher is better), which focuses on structural similarities between images:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6.21}$$

where $\mu_x$and μ_y are the mean value of images x and y, σ_x and σ_y are the standard deviation of images x and y, σ_{xy} is the covariance of x and y, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ (L is the dynamic range, $k_1 = 0.01$ and

[1] https://bitbucket.org/pierre_luc_st_charles/virtualptz_standalone

$k_2 = 0.03$). The values obtained from (6.21) are averaged over the three RGB channels to obtain the performance for color images. The third quantitative performance measure is the Bhattacharyya Coefficient (BC) (Bhattacharyya, 1946), which measures the closeness of the two discrete pixel probability distributions P and $\hat{P}$ corresponding to the GT and modeled images:

$$BC = \sum_{j=0}^{255} P(j)\hat{P}(j) \tag{6.22}$$

where $BC \in [0, 1]$ and higher is better.

In addition, Central Processing Unit (CPU) times were measured offline for each input binary file to obtain the real running time of each algorithm per frame, without including the extra waiting time to acquire the next frame that would be required if we did it online.

6.3.2 Sequences

For the experiments we used two video databases. One of them has three videos which are available on the virtualptz website. They are three indoor sequences, two of them with the same scene, and they are named *scenario3*, *scenario4* and *scenario5*. Because the two first ones are very similar, we have just shown the image results of the *scenario3* (3500x1750 pixels and 566 video frames). The video *scenario5* (3500x1750 pixels and 1957 frames) shows a room with people moving on in and doing different actions. On the other hand, we used one video sequence of the Littlstar web page[2]. We named this video as *scenario6* (2880x1440 pixels and 1169 frames) and shows a beach with people moving and playing beach voleyball. Statistics were calculated using the four sequences.

6.3.3 Parameter selection

A set of tuned parameters is needed to define the competitive learning model. These fixed parameters are reported in Table 6.1.

In addition, a study of the number of neurons has been done and the results are displayed in Figure 6.2. As seen, the equilibrium state is reached quickly and excellent results are obtained from only 612500 neurons (10% of total pixels).

6.3.4 Competitors

We have compared our proposed method with other three methods. The competing methods read each frame, then position it in the panoramic image matrix and finally compute the mean over all the frames.

[2] https://littlstar.com/

Table 6.1: Considered parameter values for our competitive learning method

Parameter	Value
Number of neurons N	$(Number\ of\ pixels)/2$
Initial learning rate η_I	0.4
Convergence learning rate η_C	0.01
Number of steps n	1000

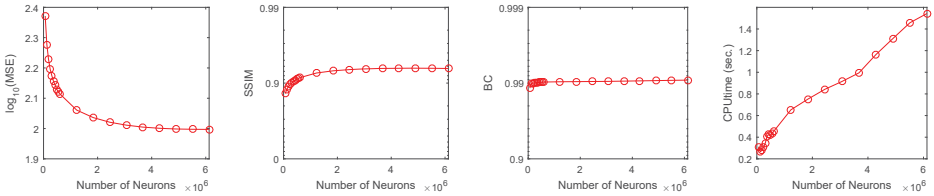


Figure 6.2: Evolution of the MSE, SSIM, BC and CPU time varying the number of neurons. *scenario3* was used for the analysis. MSE has been displayed in a logarithmic scale (lower is better), SSIM and BC in a reverse logarithmic scale (higher is better).

Since data input coordinates are almost always fractional numbers, the panoramic integer coordinates where the incoming frame must be placed were calculated by rounding the original fractional coordinates. Besides, the coordinate pairs do not define a regular strictly monotonic grid, i.e., the points have no structure or order between their relative locations, so that the usual interpolation methods defined to obtain the interpolated RGB values in a panorama cannot be used. Therefore, the scattered interpolant method from MATLAB was employed to manage this situation. It provides the following interpolation variants: 'nearest' (nearest-neighbor interpolation), 'linear' (linear interpolation), and 'natural' (natural-neighbor interpolation). The same input data was provided to all the competitors as well as our method, i.e. the 3000 camera frames saved in binary files.

6.3.5 Results

A comparison of all the evaluated methods for one of the selected videos is reported in Figure 6.3. This figure shows the comparative evolution of the methods with respect to the number of captured PTZ camera frames, i.e. the frame index in the video sequence. It can be observed that when a certain number of frames is reached that covers almost all the panoramic image, which is around 500 frames, our method (in red) attains the best performance values, as compared to the competing methods. In particular, Mean Squared Error (MSE) enhances substantially, where lower values indicates that the obtained RGB pixel values are more precise. When the number of acquired

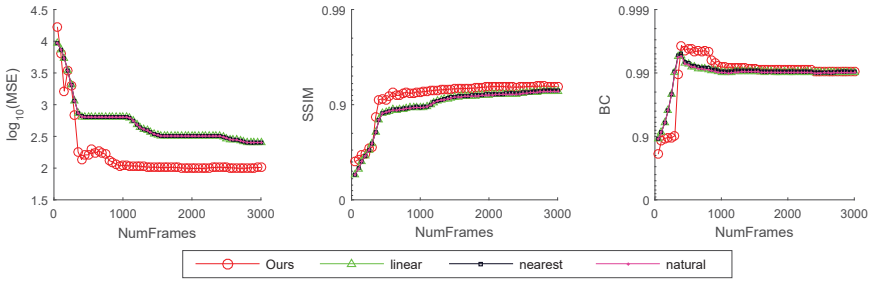


Figure 6.3: Comparative of the MSE, SSIM and BC for the four methods with *scenario3*. MSE has been displayed in a logarithmic scale (lower is better), SSIM and BC in a reverse logarithmic scale (higher is better).

Table 6.2: Mean (standard deviation) measures for each method along the four scenarios

	MSE	$SSIM$	BC	$Time(sec/frame)$
Ours	106.87 (23.09)	0.9184 (0.0371)	0.9703 (0.0466)	1.1427 (0.1284)
Linear	233.92 (35.60)	0.9167 (0.0288)	0.9701 (0.0460)	7.6549 (0.8106)
Nearest neighbor	235.00 (35.12)	0.9176 (0.0282)	0.9701 (0.0463)	7.7251 (0.8171)
Natural neighbor	234.00 (35.63)	0.9162 (0.0292)	0.9700 (0.0459)	9.1097 (1.0477)

frames increases to more than 2500, all methods tend to perform similarly, but our method still remains as the best one.

Something similar happens for the other three videos. To summarize it, we have calculated the mean and standard deviation of the four sequences for each method and for each performance measure. These qualitative results are shown in Table 6.2. As we can see, our method outperforms the competing methods, particularly in the mean squared error. Higher values of Structural Similarity Index Measure (SSIM) and Bhattacharyya Coefficient (BC) confirm that our competitive neural model produces the best approximation of the background of the scene.

The CPU time required to process one frame is a very important feature to be assessed. Table 6.2 shows the mean required time to process a binary file for a frame. Our method is around 85% faster than the interpolation methods. It computes the winner neurons and updates the quad-tree where they are stored in just over a second. If we consider that a movement of the virtual PTZ camera and the generation of the binary file for the current frame takes between one and three seconds, it turns out that our method is the only one that can be executed concurrently with the PTZ camera frame acquisition process. The competitors triplicate the required time and incur in a big time delay to obtain the panoramic background. The utilization of a GPU would improve between 25 and 50 times the processing rate.

In order to get a qualitative point of view about the suitability of our

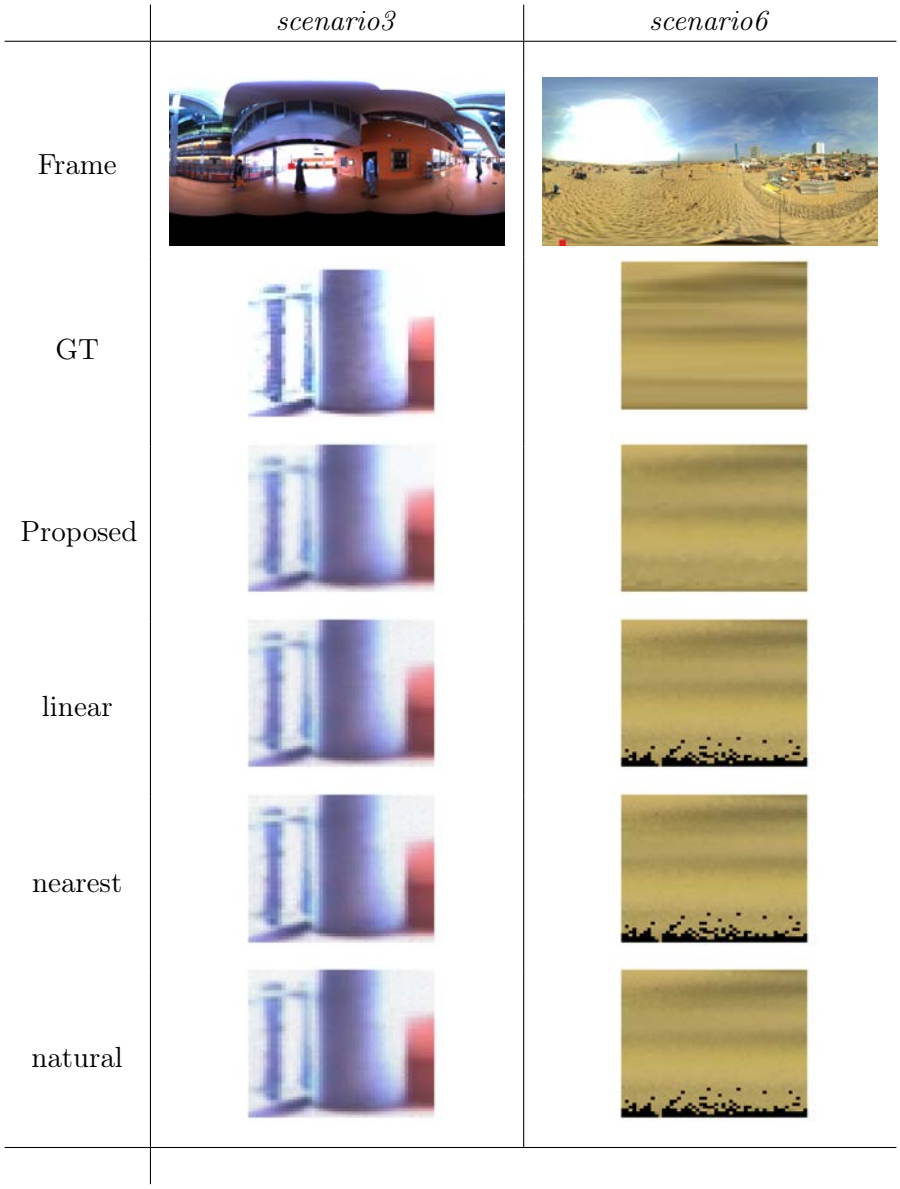


Figure 6.4: Graphical depiction of the operation of the proposed method. Each row show, from left to right, up to down: a frame of a raw panoramic video sequence, and sections (red square) of its ground truth, our algorithm, and the three competitors algorithms. Frame 153 of *scenario3* and frame 859 of *scenario6* are shown.

approach, the panoramic backgrounds generated by each method have been compared. In Figure 6.4 a window of 50x50 pixels of two scenarios are displayed, with their respective raw panoramic video frame. As it can be observed, the qualitative results offered by our approach are the most similar

Figure 6.5: Panoramic images of *scenario5* for frame 153 and the proposed method, respectively.

to the ground truth. This can be better appreciated on the top and the bottom of the panoramic image, where the competitors produce black pixels that should not be there. The window of *scenario6* is a clear example of this. Also, in the left and the right of the images produced by the competitors there are more black pixels than in our approach. With respect to the moving people of the video frame, all methods remove them efficiently and no big differences are noticed. An example of the result of our method is shown in Figure 6.5.

6.4 Conclusions

In this work, a methodology to model the panoramic background of PTZ cameras is presented. It consists of an online learning method based on a competitive neural network to read each camera frame and process it to generate a panoramic image of the scene. Four scenes have been tested to check the feasibility of the system, obtaining suitable and successful results. Also, it has been demonstrated that our approach outperforms several competing methods. It is remarkable that the proposed model uses very little CPU time to process each input video frame, which permits a easy integration in a real time PTZ camera video surveillance system.

Chapter 7

Panorama construction for PTZ camera surveillance with the neural gas network

Computer science really involves the same mindset, particularly artificial intelligence.

Frederick Lenz

Abstract: The construction of a model of the background of a scene still remains as a challenging task in video surveillance systems, in particular for moving cameras. This work presents a novel approach for constructing a panoramic background model based on the Neural Gas network and a subsequent piecewise linear interpolation by Delaunay triangulation. Furthermore, an ensemble model of Neural Gas networks is also proposed. The approach can handle arbitrary camera directions and zooms for a Pan-Tilt-Zoom (PTZ) camera-based surveillance system. After testing the proposed approach on several indoor sequences, the results demonstrate that the proposed methods are effective and suitable to use for real-time video surveillance applications.

7.1 Introduction

As mentioned in the previous chapter, PTZ cameras have become increasingly popular in monitoring public areas. Unlike previously proposed, in this chapter we address the problem of moving objects detection for PTZ

cameras, and propose a method using a specific type of competitive neural network. Concretely, a Neural Gas network (Martinetz and Schulten, 1991; Martinetz et al., 1993) or an ensemble of them are used to build a panoramic background model for object detection. The Neural Gas stands as a popular unsupervised learning neural network, which is commonly employed for vector quantization purposes (Dimitriadis et al., 2016; Antonakakis et al., 2016). As the input data is quite large, this proposal also requires a large number of neuron prototypes. Thus, quad-trees are used again to organize them quickly and efficiently.

The rest of the chapter is organized as follows. In section 7.2, a more detailed description of the proposed neural models are presented. In section 7.3, we present the results achieved with the implementation of the proposed approach. Finally, section 7.4 includes some concluding remarks.

7.2 The models

In this section a Neural Gas network based system to learn the background of a panoramic scene from the input of a PTZ camera is designed. Furthermore, an ensemble model composed of several Neural Gas networks is also proposed. The Neural Gas model is described in Subsection 7.2.1. After that, the ensemble model is proposed (Subsection 7.2.2). The data acquisition procedure as well as the interpolation technique are the same as the ones described in the previous chapter (Subsections 6.2.1 and 6.2.3).

7.2.1 Neural Gas network

Next a Neural Gas model (Martinetz and Schulten, 1991; Martinetz et al., 1993) for panoramic scenes captured by PTZ cameras is developed. In order to learn the details of the scene, a large number of neurons N is employed, which is lower but in the same order of magnitude as the number of pixels of the full panorama. Following the strategy in (Luque-Baena et al., 2015), the input vectors $\mathbf{x} \in \mathbb{R}^5$ are divided into two sections. The first section contains the positional information in the video frame, while the second section contains the color features. In our case, the first section comprises the two spherical coordinates ϕ and θ, while the second section contains the RGB color values. All the components of the input vectors are used to update the neuron prototypes $\mathbf{w}_i \in \mathbb{R}^5$, but only the two first ones participate in the competition. This way, only the positional information is employed to determine which neurons are closest to the input vector. Therefore, each neuron represents the average color in a receptive field which is one region of a Voronoi tesellation of the full panorama. That is, the neurons specialize on small pixel neighborhoods of the panorama. Since the positional components can have fractional values, $(x_1, x_2) \in \mathbb{R}^2$, the network can learn fractional

positional information without having to round to integer pixel positions in the panorama, which avoids losing valuable information.

Each neuron contains a prototype $\mathbf{w}_i \in \mathbb{R}^5$ and also a Boolean flag $b_i \in \{true, false\}$. The flag is required to control the initialization of the neuron. Since the PTZ camera does not cover the entire panorama at a time, it is not possible to initialize all the neurons at the same time. They can only be initialized while their positional components fall into the current field of view. To this end, the Boolean flags are initialized to false. The first time that a neuron wins, its prototype is set to the input sample, and its flag is set to true. From that point on, the neuron will be updated according to the Neural Gas learning rule.

The proposed learning algorithm is as follows:

1. Draw a training sample $\mathbf{x}$ at random from from the current episode $\mathcal{S}_t$, which has not been considered before, i.e. a random sampling without replacement is done.

2. For each neuron i compute the number k_i of neurons such that they are closer than i to the input vector in terms of Euclidean distance according to the first two vector components:

$$k_i = \sum_{j \in \{1,\dots,N\}} \mathbb{I}\left(d_j < d_i\right) \tag{7.1}$$

$$\forall j \in \{1, ..., N\}\,,\; d_j = \|(w_{j,1}, w_{j,2}) - (x_1, x_2)\| \tag{7.2}$$

$$\mathbb{I}\left(condition\right) = \begin{cases} 0 & \text{if } condition \text{ is false} \\ 1 & \text{if } condition \text{ is true} \end{cases} \tag{7.3}$$

   where $\mathbb{I}$ stands for the indicator function.

3. Let q be the index of the closest (winning) neuron, $k_q = 0$. If the flag b_q is true, then go to step 4. Otherwise, set the prototype of the winning neuron q to the training sample and set its flag to true:

$$\mathbf{w}_q = \mathbf{x} \tag{7.4}$$

$$b_q = true \tag{7.5}$$

   Then go to step 4.

4. Update the neuron prototypes for the M closest neurons according to the Neural Gas learning rule:

$$\mathcal{N} = \{i \in \{1, ..., N\} \mid k_i < M\} \tag{7.6}$$

$$\forall i \in \mathcal{N},\ \Delta \mathbf{w}_i = \epsilon \exp\left(-\frac{k_i}{\lambda}\right)(\mathbf{x} - \mathbf{w}_i) \tag{7.7}$$

$$\forall i \notin \mathcal{N},\ \Delta \mathbf{w}_i = 0 \tag{7.8}$$

where ϵ is a decaying learning rate and λ is a decaying neighborhood radius.

5. If all the samples of the current episode $\mathcal{S}_t$ have already been processed, then go to step 6. Otherwise, go to step 1.

6. If the last time instant t has been reached, then stop. Otherwise, increment the time instant counter t, load the next episode and go to step 1.

There are two phases in the learning process: first the ordering phase where ϵ experiences a linear decay (initial learning rate ϵ_I); and then the convergence phase where ϵ remains constant at a small value (ϵ_C). This is because the ordering phase is required for the warm-up of the algorithm only, and after that the system runs for an indefinitely long time. The change of phase is monitored by a step parameter n.

$$\epsilon(t) = \begin{cases} \epsilon_I(1 - t/n), \text{ if } t < n \\ \epsilon_C, \text{ otherwise} \end{cases} \tag{7.9}$$

λ also experiences a linear decay as ϵ with the following equation.

$$\lambda(t) = \begin{cases} \lambda_I - (\lambda_I - \lambda_C)(1 - t/n), \text{ if } t < n \\ \lambda_C, \text{ otherwise} \end{cases} \tag{7.10}$$

A theoretical analysis of the above algorithm can be carried out. Since the competition is done on the first two components, the algorithm seeks a local minimum of an energy function $\mathcal{E}$ (Martinetz et al., 1993), which only takes into account these components:

$$\mathcal{E} = \frac{1}{2C(\lambda)} \sum_{i \in \{1,\ldots,N\}} \int d_i^2 \exp\left(-\frac{k_i}{\lambda}\right) p(x_1, x_2)\, dx_1 dx_2 \tag{7.11}$$

$$C(\lambda) = \sum_{k=0}^{M-1} \exp\left(-\frac{k}{\lambda}\right) \tag{7.12}$$

where F_i is the receptive field of the i-th neuron:

$$F_q = \left\{ \mathbf{x} \mid q = \arg \min_{i \in \{1,\ldots,N\}} d_i \right\} \tag{7.13}$$

On the other hand, the prototype update is carried out on the last three components too, so those components approximate the average color of the receptive field:

$$(w_{i,3}, w_{i,4}, w_{i,5}) \approx E\left[(x_3, x_4, x_5) \mid F_i\right] \tag{7.14}$$

As a last remark, it must be pointed out that the large number of neurons N to be used for this application requires a considerable optimization of the competition equation (7.1). This is accomplished by inserting all the first sections $(w_{i,1}, w_{i,2})$ of the neuron prototypes into a quad-tree (Frisken and Perry, 2005). A quad-tree is a data structure based on a recursive decomposition in order to represent the information. It is suitable for image processing because of their properties to consider the subsets of the data which are more interesting, obtaining an enhanced execution time, so that, improving the performance of the approach where it is applied (Samet, 1984). This way (7.1) is evaluated very quickly, even for values of N in the millions.

7.2.2 Ensemble model

As mentioned before, the number of neurons N required for this application is in the millions. In order to reduce the time required to update the network, so that faster training times are attained, an ensemble model is proposed next.

An ensemble of Q Neural Gas networks will be used, each with N neurons. The members of the ensemble operate in parallel, so that each network randomly draws an input vector from the input dataset independently from the other networks. This way, a grand total of QN neurons are trained, while the overall training time can be close to that of a single network with N neurons, provided that at least Q parallel processors are available.

Once the training is finished, a unified network made of QN neurons is built by joining the neurons coming from the Q members of the ensemble. This way, a more accurate representation of the input distribution is obtained with reduced training time requirements.

7.3 Experimental results

In this section we report the computational experiments we have carried out and their results. The software and hardware that have been used are specified in Subsection 7.3.1. Then, the tested video sequences are described in 7.3.2. The descriptions of the used parameters and the proposed methods are in Subsection 7.3.3, and the descriptions of the competitors in 7.3.4. Finally, the obtained results from the experiments are reported in Subsection 7.3.5.

7.3.1 Methods

The camera control module is based on the *virtualptz library* (Chen et al., 2015). It simulates a PTZ camera from a panoramic video sequence, and it is accessible from its website [1]. The implementation is written in C++ and it uses the OpenCV and OpenGL libraries. This virtual camera has limitations in its vertical movement, going from 0 (up) to 180 (down) degrees.

To generate the input data for the Neural Gas network and their the competitors, a exhaustive scanning of the scene has been carried out. To simulate the real behavior of a PTZ camera in a practical setting, some limitations in the scan have been imposed. Starting at the top (vertical 0 degrees), we turn 360 degrees to the right with a step of 10 degrees, and when we come back to the initial point, we go down 10 degrees. This process is repeated until we arrive to the bottom and we start to go up again. Furthermore, a random zoom (vertical field of view) is applied to each frame, but again, we try to simulate a real situation. To this end, we generate a random number in an interval $[FOV, FOV + 5]$ and move this interval by steps of 5 degrees between a minimum and maximum value for the field of view, $[70, 140]$ degrees, in order to avoid strange images with irregularities.

A total of 3000 camera frames have been saved in binary files to read them synchronously by all the methods, which have been implemented in MATLAB R2015b. If the panoramic video sequences has less than 3000 frames, then we cause a loop in the video to obtain the remaining frames. The reported experiments have been carried out on a 64-bit Personal Computer with an eight-core Intel i7 3.60GHz CPU, 32 GB RAM and standard hardware. The implementation of our approach does not use any GPU resources.

Panoramic Ground Truth (GT) has been calculated doing the median of the raw panoramic video frames, but only for a maximum of 600 frames since it is necessary a considerable amount of memory to process it. This is not very relevant because we are going to compare all methods with the same image, so it does not matter if objects that do not belong to the real background appear in the obtained GT. For each method, we compared the panoramic image obtained with the ground truth. Two quality measures were used to evaluate the proposed approach: the first was the Mean Squared Error (MSE) metric (lower is better), which is commonly used in image and video processing; the second was the Structural Similarity index (higher is better), which focuses on structural similarities between images:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{7.15}$$

where $\mu_x$and μ_y are the mean value of images x and y, σ_x and σ_y are the standard deviation of images x and y, σ_{xy} is the covariance of x and y,

[1] https://bitbucket.org/pierre_luc_st_charles/virtualptz_standalone

$c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ (L is the dynamic range, $k_1 = 0.01$ and $k_2 = 0.03$). The values obtained from (7.15) are averaged over the three RGB channels to obtain the performance for color images. Last, CPU times were measured offline for each input binary file to obtain the real running time of each algorithm per frame, without including the extra waiting time to acquire the next frame that would be required if we did it online.

7.3.2 Sequences

For the experiments, a total of 6 different sequences obtained from two video databases are utilized. One of them has three videos which are available on the *virtualptz* library website. They are three indoor sequences, but two of them with the same scene, and they are named *scenario3* and *scenario4*, so we have only selected the first one for the final experiments. The second video that we used from this database is *scenario5*. The *scenario3* (3500x1750 pixels and 566 video frames) video shows a shopping mall and *scenario5* (3500x1750 pixels and 1957 frames) shows a room, both of them with people moving on in and doing different actions. On the other hand, we used four video sequences of the Littlstar web page[2]. We named these videos as *scenario6* (2880x1440 pixels and 1169 frames), which shows a beach with people moving and playing beach volleyball, *scenario7* (2048x1024 pixels and 1380 frames), which shows a train station with people, *scenario8* (1920x960 pixels and more than 7800 frames), which shows a big beach with some cars and people passing in the left side of the image, and *scenario9* (1920x960 pixels and more than 9000 frames), which shows a shore near a pier. Statistics were calculated using the six sequences. An important detail to consider is that the four videos from Littlstar have a dynamic background, like movements produced by the air in the train station, or the waves in the sea, and also the sun over time.

7.3.3 Parameter selection and proposals

A set of tuned parameters is needed to define the neural gas model. This study has been done using *scenario4*, which is different of the other videos used for the final experiments. These fixed parameters are reported in Table 7.1. We have based ourselves on the parameters presented in articles as Martinetz and Schulten (1991) and Martinetz et al. (1993) and we have tried to find a balance between quality and time efficiency (which depends on the value of M).

Taking into account the preliminary tests, we propose four different versions of the Neural Gas network varying the number of neurons: NG30, NG50, RENG30 and RENG50. The first two consist on the simple Neural

[2] https://littlstar.com/

Table 7.1: Considered parameter values for our Neural Gas network

Parameter	Value
Number of steps n	1000
Initial learning rate ϵ_I	0.4
Convergence learning rate ϵ_C	0.01
Number of closest neurons M	2
Initial neighborhood radius λ_I	1.0
Convergence neighborhood radius λ_C	0.01

Gas network using 30% and 50% of neurons with respect to the total number of pixel of the panoramic image. We have selected a thirty percent to show that we can obtain almost the same results that with fifty percent of neurons. In addition, we tested two ensemble models with the same quantity of neurons, called RENG.

7.3.4 Competitors

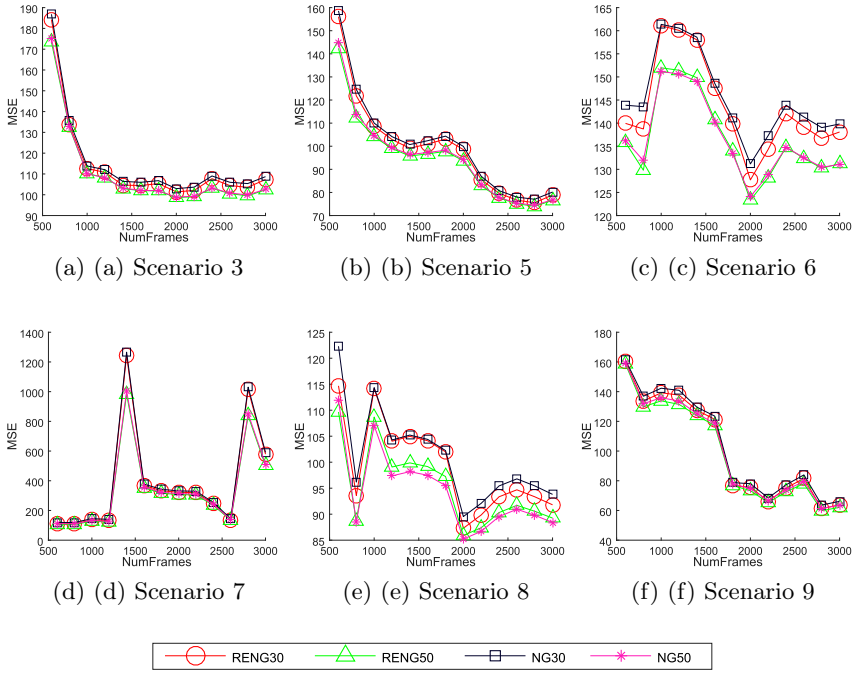
We have compared the four proposed methods based on Neural Gas network to three other methods. The competing methods read each frame, then position it in the panoramic image matrix and finally compute the mean over all the frames.

Since data input coordinates are almost always fractional numbers, the panoramic integer coordinates where the incoming frame must be placed were calculated by rounding the original fractional coordinates. Besides, the coordinate pairs do not define a regular strictly monotonic grid, i.e., the points have no structure or order between their relative locations, so that the usual interpolation methods defined to obtain the interpolated RGB values in a panorama cannot be used. Therefore, the scattered interpolant method from MATLAB was employed to manage this situation. It provides the following interpolation variants: 'nearest' (nearest-neighbor interpolation), 'linear' (linear interpolation), and 'natural' (natural-neighbor interpolation). The same input data was provided to all the competitors as well as our method, i.e. the 3000 camera frames saved in binary files.

7.3.5 Results

A comparison from a quantitative point of view between the results produced by each method can be observed in Figures 7.1, 7.2, 7.3 and 7.4.

Figure 7.1 shows the MSE of the four proposed configurations of the Neural Gas model. This figure displays the comparative evolution of the methods with respect to the number of captured PTZ camera frames, i.e. the frame index in the video sequence. For all the scenarios, the best method



(a) (a) Scenario 3 (b) (b) Scenario 5 (c) (c) Scenario 6

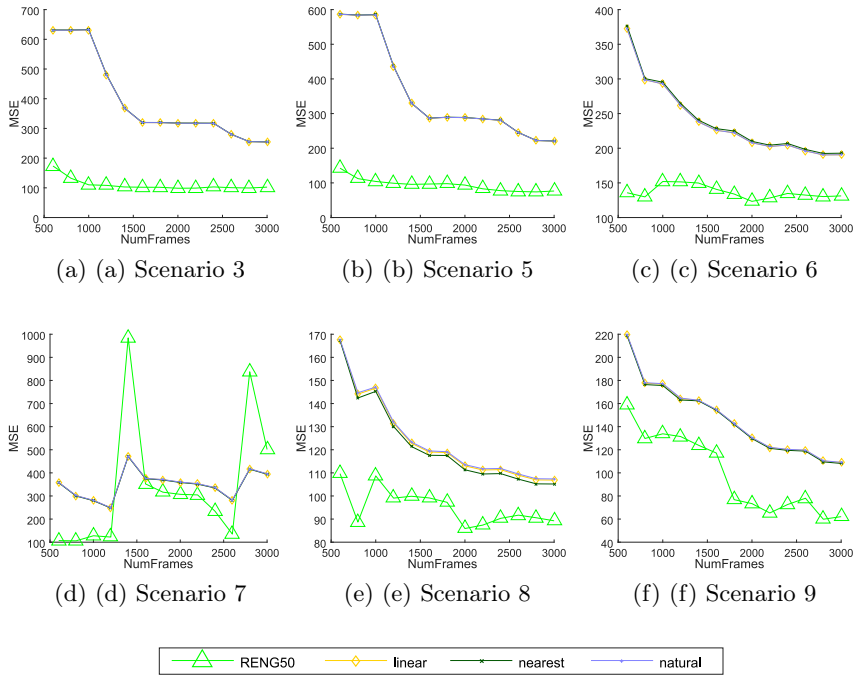
(d) (d) Scenario 7 (e) (e) Scenario 8 (f) (f) Scenario 9

(g)

Figure 7.1: Comparison of the MSE (lower is better) for the four methods proposed in this research.

is RENG50, followed by NG50 (except for *scenario8*, where both roles are exchanged). The second observation is that there is no significant differences between the four methods in most scenarios. For example, RENG30 obtains results close to NG50, which means that we can use a lower quantity of neurons, i.e., a lower processing time, to obtain the same results. The tendency shows that from 2000 frames, the MSE decrease and do not change a lot in the rest of the processing.

The comparison with the competitors is presented in Figure 7.2. We only draw the best of our methods (RENG50) for the sake of clarity. When a certain number of frames is reached that covers almost all the panoramic image, which is around 600 frames, our method (in green) attains the best performance values, as compared to the competing methods. In particular, lower values of MSE indicates that the obtained RGB pixel values are more precise. The Neural Gas model do not need a lot of frames to reach a good performance, in contradistinction to the interpolation methods, which need 1500 frames at least. When the number of acquired frames increases to more than 2500, all methods tend to perform similarly, but our method still remains as the best one.



(a) (a) Scenario 3 (b) (b) Scenario 5 (c) (c) Scenario 6

(d) (d) Scenario 7 (e) (e) Scenario 8 (f) (f) Scenario 9

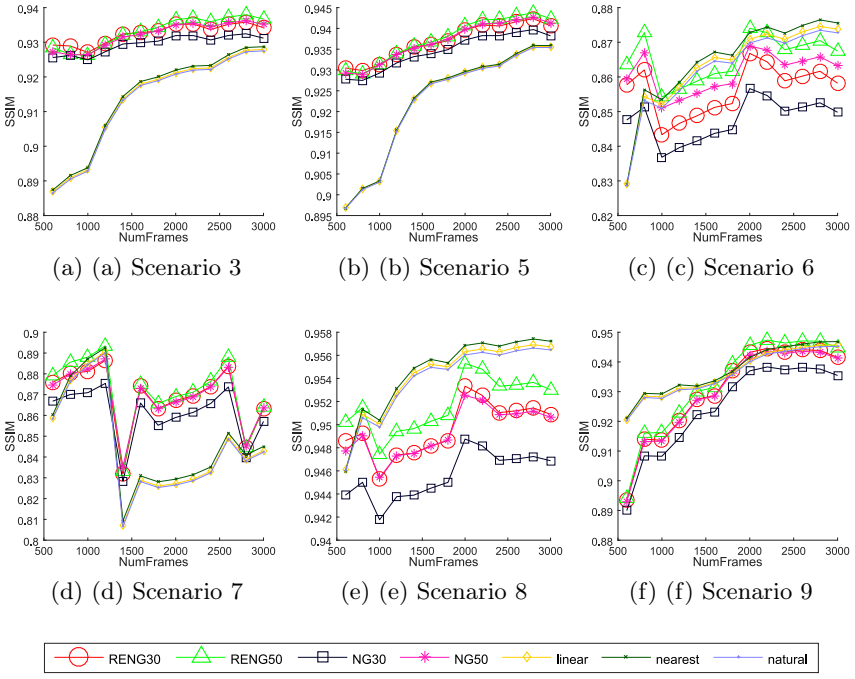
(g)

Figure 7.2: Comparison of the MSE (lower is better) for the RENG50 method and the competitors.

SSIM results for the seven methods is displayed in Figure 7.3. For the videos with fixed background, i.e., *scenario3* and *scenario5*, the Neural Gas model clearly overcomes the performance of the competing methods. Among our methods, RENG50 still remains the best. However, when an external factor is present, like wind or sun, i.e, a dynamic background, the final result is affected. In the three scenarios showing a beach, the competitors achieve a better final panoramic background, but the difference with respect to our proposal is less than 1%. If we also consider the MSE to perform a global evaluation, it is clear that the neural model is more suitable.

There is a particularity in the results of *scenario7* (Figures 7.1d, 7.2d and 7.3d). We can appreciate two peaks around 1400 and 2800 frames. That is because this video has a duration of 1 min. 32 s., it has 1380 video frames, and in the last seventy frames, a man cover the camera with a cardboard, so the neural network treat to learn it and the comparison with the GT generates bad results. But when more frames are processed, the net has the ability to recover very fast and obtain better results than the competitors.

Last, it is important to remark that the evaluation for *scenario9* has been carried out comparing with two different ground truths. This video



(a) (a) Scenario 3 (b) (b) Scenario 5 (c) (c) Scenario 6

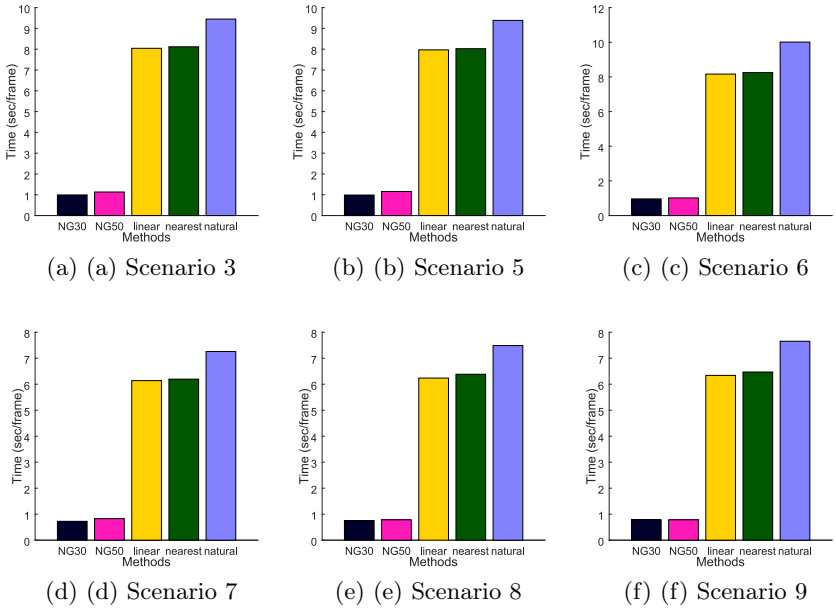
(d) (d) Scenario 7 (e) (e) Scenario 8 (f) (f) Scenario 9

(g)

Figure 7.3: Comparison of the SSIM (higher is better) for the the seven methods.

has a particularity: in the first 1350 frames approximately, there are not many people walking around, but then two men arrive and sit on the shore, and stay there until the end off the video. Our model finishes learning this situation but the competitors have it more smoothed, as they do the average of all the frames. When compared to the ground truth, they do a little better. Therefore, we generate another GT taking the six hundred frames to the previously mentioned, and we compared all the methods with both GTs. The original one for the 1349 frames and the second one for the rest. The results were combined in the same graphic (Figures 7.1f, 7.2f and 7.3f). We can observe that our methods improve the competitors in the MSE, and equal the SSIM in the case of RENG50.

A bar graph is shown in Figure 7.4 to appreciate the large differences in time required to process a frame. Mean values for the 3000 frames are calculated. Since Random Ensemble methods (RENG) would require almost the same time for the calculations as their simple version (NG), only the latter has been shown. It is clear that our proposals achieve a better performance than the competing methods for any scenario. The similarity between NG30 and NG50 and the so low computing time would permit to vary the



(a) (a) Scenario 3 (b) (b) Scenario 5 (c) (c) Scenario 6

(d) (d) Scenario 7 (e) (e) Scenario 8 (f) (f) Scenario 9

Figure 7.4: Comparison of the average CPU time required to train the 3000 frames captured by the virtual PTZ camera. Only five methods are shown because Random Ensemble methods have the same processing time as the simple method.

number of neurons adapting to each type of scenario to try to find the best configuration.

To summarize, we have calculated the mean and standard deviation of the six sequences for each method and for each performance measure. These qualitative results are shown in Table 7.2. As we can see, the Neural Gas model clearly outperforms the competing methods in terms of the mean squared error. Higher values of SSIM confirm that our neural model produces the best approximation of the background of the scene, with special remark to the RENG50 method.

The CPU time required to process one frame is a very important feature to be assessed. Table 7.2 shows the mean required time to process a binary file for a frame. Our proposed methods are around 87% faster than the best of the interpolation methods. It computes the winner neurons and updates the quad-tree where they are stored in just over a second. If we consider that a movement of the virtual PTZ camera and the generation of the binary file for the current frame takes between one and three seconds, it turns out that our method is the only one that can be executed concurrently with the PTZ camera frame acquisition process. Moreover, if we have available more than one processor, we can employ the random ensemble methods RENG30 or

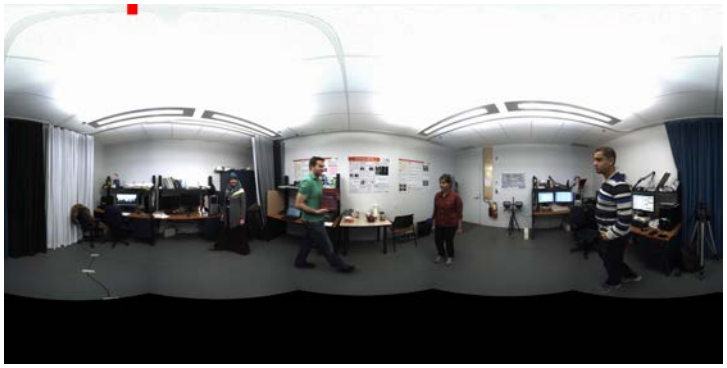
Table 7.2: Mean (standard deviation) measures for each method along the six scenarios. (*) CPU time for these methods has been established the same as their simple version method because they can be parallelized.

Method	MSE	$SSIM$	$Time(sec/frame)$
NG30	179.9321 (202.9430)	0.9098 (0.0440)	**0.9211** (**0.1359**)
NG50	162.0514 (171.9265)	0.9159 (0.0409)	1.0281 (0.1876)
RENG30 (*)	176.4957 (198.8611)	0.9148 (0.0423)	0.9211 (0.1359)
RENG50 (*)	**160.4683** (168.6281)	**0.9180** (**0.0409**)	1.0281 (0.1876)
Linear	212.7255 (**107.3339**)	0.9138 (0.0451)	7.0936 (1.0673)
Natural neighbor	212.9676 (107.3528)	0.9133 (0.0453)	8.4727 (1.2829)
Nearest neighbor	212.6441 (107.3512)	0.9149 (0.0444)	7.1806 (1.0532)

Figure 7.5: Panoramic images of *scenario7* for frame 300 and the RENG50 proposed method, respectively.

RENG50 to improve the quality with the same time. The competitors require seven times more time and incur in a big time delay to obtain the panoramic background. The utilization of a GPU would improve between 25 and 50 times the processing rate, since the inner loop of the proposed method is amenable to single instruction multiple data (SIMD) parallelization. Therefore, each pixel can be processed by a GPU thread. Present day GPUs can run hundreds or thousands of such threads at a time. Given the single thread CPU times shown in Figure 7.1, the number of frames that can be processed on a GPU could be increased to 50 frames per second, thereby achieving real time performance.

A comparison from a qualitative point of view between the results produced by each method can be observed in Figures 7.5, 7.6 and 7.7. All methods remove the moving objects of the video frame efficiently and no relevant differences are noticed. An example of the result of our method is shown in Figure 7.5, where a train passes and it does not appear in the panoramic background generated by RENG50. On the other hand, Figures 7.6 and 7.7 show two scenarios from the tested videos, where image 7.6a represents the panoramic video frame and a selected window of 50×50 pixels (marked in red), and the rest of the images represent the ground truth and the produced result by each method in that window. The qualitative results



(a) (a) Frame 588

(b) (b) GT (c) (c) Linear (220.45) (d) (d) Natural (220.59) (e) (e) Nearest (220.92)

(f) (f) NG30 (80.17) (g) (g) NG50 (76.86) (h) (h) RENG30 (79.53) (i) (i) RENG50 (76.31)

Figure 7.6: Graphical depiction of the operation of the seven methods evaluated with *scenario5*. MSE (lower is better) of the whole panoramic image is shown in parentheses.

offered by our approach are the most similar to the ground truth. This can be better appreciated on the top and the bottom of the panoramic image, where the competitors produce black pixels that should not be there. The window of *scenario5* in Figure 7.6 is a clear example of this. Also, in the left and the right of the images produced by the competitors there are more black pixels than in our approach. In addition, a better appreciation of the learned dynamic background can be observed in Figure 7.7. Our approach changes and adapts the modeled background and the new scenario with the remaining person is learned faster than the rest of the competitors.

(a) (a) Frame 1500

(b) (b) GT

(c) (c) Linear (108.91)

(d) (d) Natural (109.15)

(e) (e) Nearest (108.16)

(f) (f) NG30 (66.15)

(g) (g) NG50 (63.57)

(h) (h) RENG30 (65.38)

(i) (i) RENG50 (62.89)

Figure 7.7: Graphical depiction of the operation of the seven methods evaluated with *scenario9*. MSE (lower is better) of the whole panoramic image is shown in parentheses.

7.4 Conclusions

In this work, a methodology to model the panoramic background of PTZ cameras is presented. It consists of an online learning method based on individual or ensemble Neural Gas networks to read each camera frame and process it to generate a panoramic image of the scene. Six scenes have been tested to check the feasibility of the system, obtaining suitable and successful results. Also, it has been demonstrated that our approaches outperform several competing methods. It should be noted that the a reduction in the number of neurons (from 50% to 30% of the pixels) implies a slight decrease of the performance metrics, although visually it is not too evident. Additionally, the ensemble models (RENG30 and RENG50) help to improve the background modeling, raising the performance rates. Finally, it is remark-

able that the proposed models uses very little CPU time to process each input video frame, which permits an easy integration in a real time PTZ camera video surveillance system.

Chapter 8

Diabetic wounds segmentation using convolutional neural networks

Make everything as simple as possible,
but not simpler.
Albert Einstein

Abstract: Image segmentation is a permanent goal in many medical applications, in order to keep improving diagnostic capability and healing prediction. In order to assess the wound healing rate in diabetic foot ulcers, some parameters from the wound area are measured. However, heterogeneity of diabetic skin lesions and the noise present in images captured by digital cameras make wound extraction a difficult task. In this work, a Deep Learning based method for accurate segmentation of wound regions is proposed. In the proposed method, input images are first processed to remove artifacts and then fed into a Convolutional Neural Network (CNN) producing a probability map. Finally, the probability maps are processed to extract the wound region. Moreover, we address the problem of removing some false positives. Experiments show that our proposed method can achieve high performance in terms of segmentation accuracy and Dice index.

8.1 Introduction

Deep Learning (DL) methods have been applied in several applications in the fields of image processing, computer vision and pattern recognition. Application of DL methods on medical images has led to a great advance in

detection, classification and prediction of many types of diseases (Schmidhuber, 2015; Litjens et al., 2017). There are recent examples where a CNN is applied for image segmentation in medical applications, such as multi-organ segmentation on abdominal CT images or multiclass brain tumor segmentation on magnetic resonance images (Chen et al., 2018a). Another interesting application of CNN on skin lesions is described in (Pal et al., 2018), where the objective is the segmentation of psoriasis lesions affecting human skin biopsy.

Images of diabetic skin lesions are more challenging due to color and texture heterogeneity, as well as illumination variations, reflection and presence of noise. In (Babu et al., 2018), Subudhi *et al.* proposed a segmentation method based on k-means clustering to segment diabetic wounds. An automatic tissue identification in wound images was proposed by Veredas *et al.* (Veredas et al., 2010), where color and texture patterns are fed into a set of multilayer perceptrons. Another framework for automatic wound segmentation and healing prediction is proposed in (Wang et al., 2015) where a kind of auto-encoder architecture called ConvNet is used for segmentation.

In this paper, we propose a CNN-based method for extraction of diabetic lesion regions from skin images. We include the well-known U-Net deep network for image segmentation (Ronneberger et al., 2015). We also adapt the CNN architecture of previous works based on melanoma detection (Jafari et al., 2016, 2017) to segment diabetic wound lesions. The novelty is that we design a simplified model to process a wide range of diabetic skin wounds, incorporating an adaptive thresholding method to remove artifacts. We also perform a pre-processing step to homogenize the input, hence the images are captured by standard cameras; and include some post-processing operations to remove noisy artifacts and to select the correct wound regions based on mean probabilities of the connected components. To have a better evaluation of the performance of our model, we also design an SVM binary classification model for image segmentation, which has been successfully used in classification and regression problems, and compare the outputs of our CNN model with the SVM model.

The rest of this chapter is organized as follows. In Section 8.2, the proposed model is presented and both pre-processing and post-processing steps are explained in detail. Comparative results are reported in Section 8.3, where the dataset and the parameter selection procedure are also described. Finally, the conclusions and future works are presented in Section 8.4.

8.2 Proposed model

In this section the wound segmentation model is explained in detail. The schematic diagram of the proposed method is depicted in Fig. 8.1.

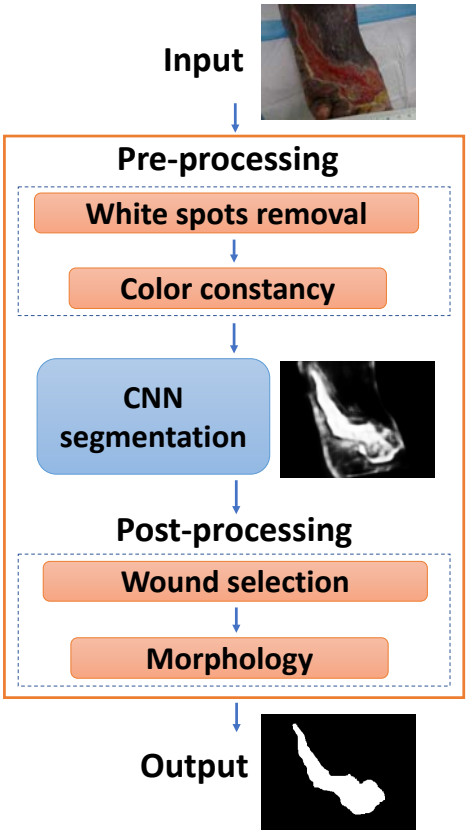


Figure 8.1: Scheme of the proposed algorithm.

8.2.1 Pre-processing

The photographs of the wounds are heterogeneous in terms of illumination, dimension, artifacts and other characteristics. Illumination is one of the main issues that must be considered in pre-processing. Therefore, we first reduce the noise effect of white spots caused by light reflection by the application of a modified version of a mean filter. A threshold, σ, is used to select those pixels with the highest brightness. Next, the following kernel is generated and convoluted over all the selected pixels:

$$K = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{8.1}$$

In this way, the white spots are substituted by the mean value from all neighbor pixels, without taking into account the high value of the processed white pixels.

Moreover, color constancy is used to normalize image color to eliminate the influence of different lighting situation (Barata et al., 2015). Shades of

gray (van de Weijer et al., 2007) can estimate the color of light source for a color image $\mathbf{I}$,

$$e_c = \frac{1}{k}\left(\frac{\int(\mathbf{I}_c(x))^p d\mathbf{x}}{\int d\mathbf{x}}\right)^{1/p} \tag{8.2}$$

where e_c, $c \in \{R, G, B\}$, is the component of the color of light source in the c-th channel, $\mathbf{I}_c$ is the component of image $\mathbf{I}$ in the c-th channel, $\mathbf{x}$ is the position of a pixel, and k is the normalization constant to ensure $\mathbf{e} = [e_R, e_G, e_B]$ to be a unit vector. Moreover, p is a norm degree. Then, we can get the coefficient d_c following the von Kries diagonal model (von Kries, 1970) to transform the image I as follows:

$$d_c = \frac{1}{\sqrt{3}e_c}, c \in \{R, G, B\} \tag{8.3}$$

$$\begin{pmatrix} \mathbf{I}'_R \\ \mathbf{I}'_G \\ \mathbf{I}'_B \end{pmatrix} = \begin{pmatrix} d_R & 0 & 0 \\ 0 & d_G & 0 \\ 0 & 0 & d_B \end{pmatrix} \begin{pmatrix} \mathbf{I}_R \\ \mathbf{I}_G \\ \mathbf{I}_B \end{pmatrix} \tag{8.4}$$

where $[\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B]^T$ denotes the intensity of an image acquired in the unknown light source, and $[\mathbf{I}'_R, \mathbf{I}'_G, \mathbf{I}'_B]^T$ denotes the intensity of the transformed image.

8.2.2 Convolutional Neural Network

The CNN learns a function f that computes a probability map $\mathbf{Y}$ from an input, such as RGB image $\mathbf{X} \in [0, 255]^{h \times w \times 3}$,

$$f(\mathbf{X}) = \mathbf{Y},\ \mathbf{Y} \in [0, 1]^{h \times w} \tag{8.5}$$

where h and w represent the height and the width of the image, respectively. We also assumed that the range of pixel values is between 0 and 255. For the proposed model described in this section, we use two different CNNs to compute the segmented image. On one side, we use a patch-wise based network (Jafari et al., 2016) and extract a set of local and global patches from the input image. For each patch centered on a determined pixel, f computes the probability to be wound or non-wound region for this pixel. The local patches would focus on the texture around the central pixel, helping to determine the wound's border. The larger patch will extract more global features from the image.

The structure of the network is composed of two paths that are fed by the local and global patches. Each path $b \in \{1, 2\}$ consists of two sets of layers $g_i^{(b)}$ performing a convolution and ReLU operations:

$$g_i^{(b)} = ReLU(W_i * g_{i-1}^{(b)} + b_i),\ i \in \{1, 2\} \tag{8.6}$$

where W_i and b_i are the weights and biases of the i-th layer.

The output of the two paths of layers are connected by a dense layer and then followed by another one with two classes. Finally, a softmax layer is appended to the end to produce a probability map. Specific details of the implementation of this neural network can be found in (Jafari et al., 2016).

On the other hand, the U-Net (Ronneberger et al., 2015) is a kind of fully convolutional network that consists of a contracting path and an expansive path. The contracting path is based on a combination of convolutional and ReLU layers, arriving to multi-channel feature maps, that are reconstructed to the desired number of classes by the expansive path. A combination of 23 convolutional layers and a strong use of data augmentation achieve a very efficient deep network with few data samples.

8.2.3 Post-processing

The output of this CNN model is a probability map $\mathbf{Y} \in [0,1]^{h \times w}$ which shows how likely each pixel belongs to the wound area. The higher the intensity value is, the more likely this pixel belongs to the wound area. The goal of the post-processing is to obtain a binary image, $\tilde{\mathbf{Y}} \in \{0,1\}^{h \times w}$, that represents a mask segmentation of the source image to remove false positives as many as possible.

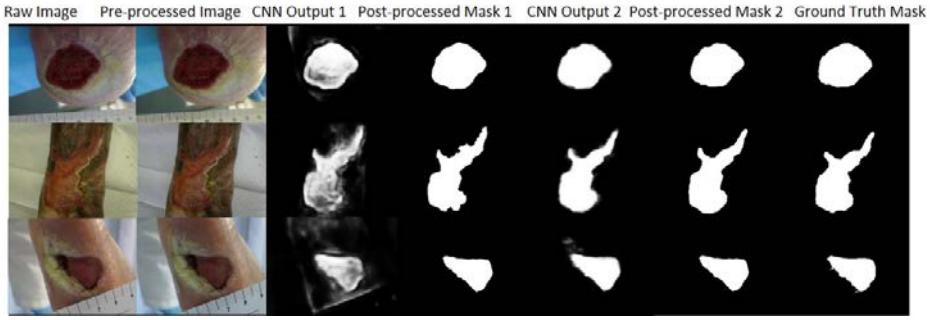


Figure 8.2: Qualitative results for three different images of the NYU dataset. From left to right, raw image, the three steps of the segmentation model and ground truth image is shown for comparison (The 3rd and 4th columns are from the two-path patch-based CNN model and the 5th and 6th columns are from the U-Net model).

1. Firstly, we use the following equation to select those connected components whose corresponding pixels in $\mathbf{Y}$ are greater than a pre-defined threshold, α.

$$S_t = \underset{n=1,...,N}{argmax}\{|\mathbf{c}^{(n)}| < |\mathbf{c}^{(S_{t-1})}|\}, t = 1, ..., N' \tag{8.7}$$

where $\{c^{(S_t)},\ t = 1, ...N'\}$ is the set of selected components and $|\cdot|$

stands for the cardinality of a set. We also assume $\{\mathbf{c}^{(n)}, n = 1, ..., N\}$ is the set of all connected components, and $|\mathbf{c}^{(S_0)}| = h \cdot w$.

2. Secondly, instead of simply keeping the largest connected component as the wound area, the mean value of the probability intensity for each selected component is computed and the connected component with the largest mean value is selected.

$$n_M = \underset{t=1,...,N'}{argmax} \left(\frac{1}{|\mathbf{c}^{(S_t)}|} \sum_{(i,j) \in c^{(S_t)}} \mathbf{Y}(i,j) \right) \tag{8.8}$$

3. Thirdly, to reduce the false positive selection, two rules are added to discard the selected connected components which are less likely to be wound.

   - The center of the selected component $\mathbf{c}^{(n_M)}$ is close to the border of the image.
   - The area of the selected component $\mathbf{c}^{(n_M)}$ is too small compared with the wound with the largest connected component:

$$|\mathbf{c}^{(n_M)}| < \tau \cdot \underset{t=1,...,N'}{max} \left(|\mathbf{c}^{(S_t)}| \right) \tag{8.9}$$

     where $\tau \in (0, 1)$ is a fixed threshold to remove small components, even if the mean probability of the component is very high.

If the selected connected component does not satisfy the rules of step 3, it is discarded. Then, steps 2 and 3 are repeated with the rest of the $N' - 1$ regions until the two rules are satisfied or no components are remaining, and in this case, the component with the highest mean probability is chosen. Finally, closing, dilation, hole filling and erosion morphology operations are applied to refine the selected wound area to get the final result of segmentation.

8.3 Experimental results

8.3.1 Dataset

A dataset composed by high resolution camera photos of wound images was used for experiments, that was provided by the New York University (NYU) (Wang et al., 2015). These images were previously cropped by the GrabCut tool (Rother et al., 2004) to remove some background. The total number of images that we have is 445. Among these images, 392 images are randomly selected to do the 4-fold cross validation for hyper-parameter selection and

Table 8.1: Range of parameters tested for hyper-parameter selection of the proposed model.

Parameter	Values
$init\_lr$	$\{0.001, 0.005, 0.0001, 0.0005, 0.0001\}$
$momentum$ (μ)	$\{0.5, 0.8, 0.9, 0.95, 0.99\}$

Table 8.2: Quantitative comparison of wound segmentation models without the post-processing (higher is better).

Method	Precision	Sensitivity	Specificity	Pixel Accuracy	Mean IoU	Dice	MCC
SVM(RGB)	0.564	0.806	0.896	0.877	0.472	0.596	0.594
Patch-based CNN	0.646	0.854	0.932	0.913	0.569	0.700	0.675
U-Net	0.768	0.937	0.960	0.949	0.723	0.823	0.840

model training. Then, the trained model is tested on unseen test set with 53 images. All images have fixed size with 240x320 pixels. Since our method is patch-wise based, 4500 random pairs of local and global patches, respectively with resolutions of 31×31 and 201×201, are evenly selected from wound area, wound border and the background area on each padded image (Jafari et al., 2017). Patches are fed into the CNN to train the model.

8.3.2 Parameter selection

A 4-fold cross-validation is applied to tune the hyper-parameters of our CNN. Table 8.1 list the hyper-parameters tested for selection. In our simulations, the same hyper-parameters of every training set were selected, which are $init\_lr = 0.0001$ and $\mu = 0.99$, and the standard deviation of the pixel accuracy rate of 4 test sets were smaller than 0.7%. For the U-Net, we used the default parameters. In pre-processing, $p = 6$ is selected for shades of gray as is recommended in the work of Barata *et al.* (Barata et al., 2015). After getting the probability map generated by the CNNs, the binary threshold $\alpha = 0.5$ is selected. We also selected at most four connected components (i.e. $N' = 4$) ans set τ to be 0.125.

8.3.3 Evaluation

From a quantitative point of view, we have chosen well-known measures in order to evaluate the performance of the image segmentation. More precisely, it has been considered the Dice index and Intersection on Union (IoU), which are used to evaluating the overlapping extent of the ground truth and the

Table 8.3: Quantitative comparison of wound segmentation models with the post-processing (higher is better).

Method	Precision	Sensitivity	Specificity	Pixel Accuracy	Mean IoU	Dice	MCC
Patch-based CNN	0.722	0.9	0.947	0.934	0.660	0.770	0.753
U-Net	0.830	0.917	0.973	0.966	0.761	0.845	0.839

predicted segmentation, defined as follows.

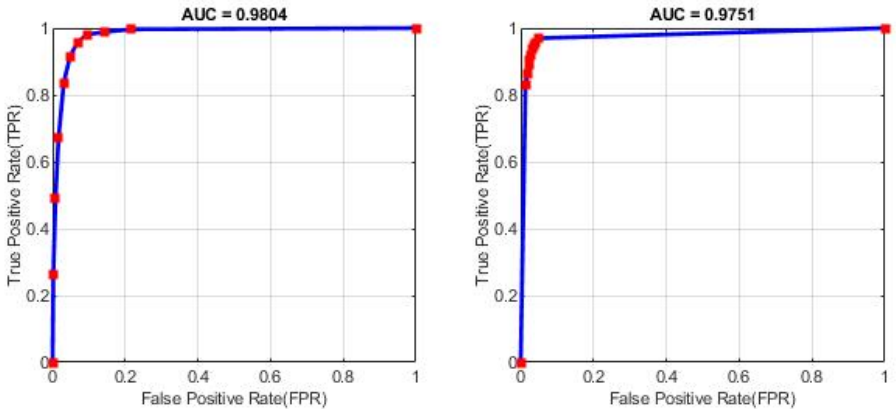
$$IoU = \frac{TP}{FN + TP + FP}$$
$$Dice = \frac{2TP}{2TP + FP + FN} \tag{8.10}$$

These measures provide values in the interval $[0, 1]$, where higher values show better performance and represent the percentage of coincidence with respect to the Ground Truth (GT). Precision, sensitivity, specificity, pixel accuracy and Matthews Correlation Coefficient (MCC) are also used in this work. $MCC \in [-1, 1]$ is more informative in evaluating binary classification problems, because it takes into account the balance ratios of the TP, TN, FP and FN. 1 represents a perfect prediction and -1 indicates complete disagreement.

To compare the performance of CNN methods, we use the linear SVM classifier with hinge loss as a baseline, treating the segmentation problem as the pixel-wise binary classification (Wang et al., 2015). For each pixel, its intensity value in the RGB channels and the surrounding 8 pixels' are extracted as its features. For training data, 1/3 of pixels belonging to the wound area are randomly selected, while 1/9 of pixels belonging to the rest area are randomly selected, in order to balance the percentage of the pixels with different labels for training.

8.3.4 Results

Fig. 8.2 presents examples of the qualitative performance of the proposed model. The pre-processing methods reduce the misleading factors by removing the white spots and adjusting to the white balanced illumination. About the CNN output, the probability map looks good since most of the background regions are vanished. The result of U-Net always have clear boundary than the result of the patch-based CNN method. To further refine the segmentation to get the predicted mask, we apply post-processing on the raw output of the CNNs. In post-processing, wound selection based on the probability value and the two rules mentioned above work together to reduce the false positive prediction, which can improve the performance of wound segmentation. Fig. 8.2 is an example where the false positive recognition is corrected based on the post-processing.



(a) patch-based CNN's (b) U-Net

Figure 8.3: ROC curves of the deep segmentation networks according to α.

The performance of the CNN models with selected hyper-parameters was run on the independent test set and results are shown in Table 8.2. Comparing the evaluation of raw segmentation with the evaluation of the segmentation with post-processing method shown in table 8.2, the proposed post-processing method works effectively in both models. In general, the U-Net model performs better than the patch-based method. For CNN methods, high values of accuracy, sensitivity and specificity are achieved, but the most indicative measures are the difference obtained in the Dice index and Matthews Correlation Coefficient (MCC), which indicates a better coincidence with the GT compared to the results obtained by the classical SVM method.

The Receiver Operating Characteristic (ROC) curve of the CNN's output according to different value of threshold α for the binary segmentation with the post-processing is shown in Fig. 8.3, reporting a high area under the curve, meaning a good performance of the proposed method. Fig. 8.4 depicts an inspection of the post-processing procedure that we carried out to remove false positives. Only applying the threshold α to the CNN's output, the 4 largest components shown in Fig. 8.4d and Fig. 8.4e would be kept. However, by computing the mean probability values these components and considering the two additional rules, our methods is able to discard the largest one shown in Fig. 8.4f, which in this example is a spurious object, and select the correct one shown in Fig. 8.4e.

8.4 Conclusions

In this paper we introduced the automatic segmentation model of diabetic wounds, consisting of pre-processing, CNN segmentation and post-processing.

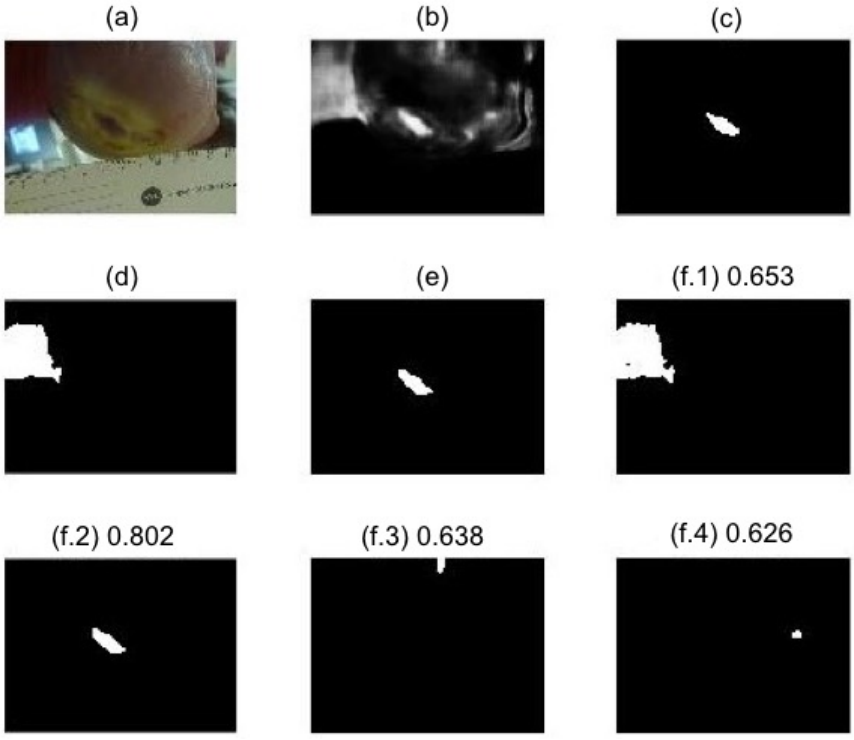


Figure 8.4: Example of the post-processing operation for the wound segmentation refinement. (a) Raw image; (b) probability map generated by the CNN model; (c) the ground-truth mask; (d) the largest connected component; (e) post-processed mask by using the proposed post-processing methods; (f.1)-(f.4) images of the 4 largest connected components kept for post-processing and the values are the average of the probability value of pixels in the corresponding component.

Color constancy and white spots removing were used to reduce the affect of different illumination. The patch-based CNN model and the U-Net model are trained to get the probability map. The post-processing based on the probability map was used to refine the segmentation results by two additional rules and the morphology method. Experiments on the NYU database demonstrated the effectiveness of the U-Net model as well as the patch based model even with less images, but the former one is more time-saving and accurate in segmentation. Probably, the future increment of the dataset would improve the model even more. The results of the automatic wound segmentation could be used for wound area measurement and the feature extraction of wounds, which will benefit a lot for the following wound diagnosis and healing processes.

Part III

Curve and Surface Fitting

Chapter 9

A fast robust geometric fitting method for parabolic curves

The unavoidable price of reliability is simplicity.

Tony Hoare

Abstract: Fitting discrete data obtained by image acquisition devices to a curve is a common task in many fields of science and engineering. In particular, the parabola is some of the most employed shape features in electrical engineering and telecommunication applications. Standard curve fitting techniques to solve this problem involve the minimization of squared errors. However, most of these procedures are sensitive to noise. Here, we propose an algorithm based on the minimization of absolute errors accompanied by a normalization of the directrix vector that leads to an improved stability of the method. This way, our proposal is substantially resilient to noisy samples in the input dataset. Experimental results demonstrate the good performance of the algorithm in terms of speed and accuracy when compared to previous approaches, both for synthetic and real data.

9.1 Introduction

Fitting of conic sections such as parabola, hyperbola and ellipse is a fundamental task in digital image analysis, pattern recognition, computer graphics, computer vision, reverse engineering and statistics. In these fields, fitting geometric primitive models is especially significant for reverse engineering, whose core is to reconstruct the curve from scattered data, and request the

fitting curve reflecting the shapes and features of the original data points (Benko et al., 2002; Conti et al., 2018; García-Escudero et al., 2017; Hu et al., 2017; Torrente et al., 2018; Várady et al., 1997; de Vieilleville and Lachaud, 2009; Xiao et al., 2016). Also, there are some records in the literature on the fitting of quadratic curves and surfaces to sampling points that occurs frequently in several metrology, astronomy, medical diagnosis, geology (Bookstein, 1979; Butler et al., 1994; G. et al., 1988; Krüger and Wöhler, 2011; van Huffel, 1997), industrial and architectural applications (Chen et al., 2017a; Samper et al., 2017). For example, a fitting of conic sections was proposed for a biological application, such as the chromosome shape analyzed by (Paton, 1970).

Most of the current state of the art methods to fit parabolas are based on the minimization of squared errors. Such methodologies have little resilience to noisy observations, due to the excessive sensitivity of the squared error function to them. This problem is further exacerbated by the fact that for many input datasets there are spurious solutions that the fitting algorithms can easily fall into. Here we aim to develop a method to fit parabolas from a set of sample points which is more robust, based on the minimization of absolute errors. The main contribution of this work is the proposal of a fitting algorithm which is resilient both to outlying input points, by means of absolute error minimization, to the presence of spurious solutions, and the introduction of random restarts in the search. Moreover, our proposal attains a good balance between the quality of the solution and the execution speed.

This chapter is structured as follows. Section 9.2 presents previous works and the definition of earlier parabola fitting methods. Subsequently, Section 9.3 describes the formulation of our method and Section 9.4 reports the obtained experimental results using synthetic and real data. Finally, Section 9.5 summarizes the main conclusions of this research.

9.2 Previous work

Many proposals can be found about parabola fitting in the literature, but there are few state-of-art methods with available source code that can be used in order to obtain interesting experimental results.

Harker (Harker et al., 2008) considers that the Direct method is the first known direct solution for parabola fitting. This fitting is reached by establishing an orthogonal basis vector set in the Grassmannian space of the quadratic terms' coefficients. The linear combination of the basis vectors that satisfied the parabolic condition and has a minimum residual norm is resolved using Lagrange multipliers.

Cals Method fits a parabola using Consistent Algebraic Least Squares (CALS). That implies a pre-processed scatter matrix for a Direct least squares

fitting. An improved matrix partitioning is employed, through an extension of Halíř and Flusser's work (Halir and Flusser, 1998). A generalization of the Eckart-Young-Mirsky matrix approximation theorem allows for an incremental orthogonal residualization of the partitioned scatter matrix (Golub et al., 1987).

García (García-Escudero et al., 2017) has presented a method to fit parabolas to scattered data which is applicable in noisy images. They employ a trimming procedure and avoids trying to fit all the points in the data set, but only a proportion $1-\alpha$ of them. This approach has the disadvantage that it does not only need the samples as input parameters, but also the trimming level α and two step parameters (L updating steps and S random initializations) for the iterative process, which depend on the characteristics of the dataset.

9.2.1 Geometric distance

The geometric distance from a point $(x, y) \in \mathbb{R}^2$ to a conic is the smallest Euclidean distance from the point to all the points in the conic (Ahn et al., 2001):

$$d_G(x, y) = \min\left\{\|(x, y) - (r, s)\| \mid (r, s) \in \mathbb{R}^2,\ Q(r, s) = 0\right\}$$

where $\|\cdot\|$ stands for the Euclidean distance.

The objective function based on minimizing the geometric distance is a 4th order equation. This can be solved by mathematical methods, but it needs to use non-linear procedures and the results are not quite stable (Sturm and Gargallo, 2007; Zhang, 1997). So the applications are somehow limited.

9.2.2 Algebraic distance

The algebraic distance from a point $(x, y) \in \mathbb{R}^2$ to a conic is given by (Zhang, 1997):

$$d_A(x, y) = Q^2(x, y)$$

The minimizing algebraic distance method has been used by several researchers. For example, (Liu et al., 2004) used six different restrictions to obtain six basic conics, and produced the final fitting conic by adding certain weights to the coefficients of the six basic conics. Rosin discussed the objective functions based on minimizing algebraic distance from the aspects of curvature bias, singularities, and transformational invariance (Rosin, 1993a). On their part, (Bookstein, 1979; Fitzgibbon et al., 1999; Rosin, 1996) introduce different constraints to the objective function, transformed it to the extreme problem, and obtained varied results depending on the restriction.

Furthermore, (Harker et al., 2008) used the partition matrix to fitting a specific conic and introduced a bias correction method. Based on the aforecited researches, a conic fitting method is provided in Li's work (Li et al., 2012), which minimizes the point-to-curve algebraic distance for the given data points. This proposed method can preserve the original outline of the conic data points, whereas the fitting effects are improved.

9.3 The model

Next our robust parabola fitting method is presented. The equation of a parabola in 2D is given by:

$$\frac{(ax+by+c)^2}{a^2+b^2} = (x-u)^2 + (y-v)^2 \tag{9.1}$$

where the directrix is $ax+by+c=0$ with a and b not both zero, and the focus point is $(u,v) \in \mathbb{R}^2$. Please note that at this point it is not advantageous to normalize the directrix vector, $a^2+b^2=1$, since such normalization would force to express either a or b in terms of the other, for example $b=\sqrt{a^2-1}$, which would clutter the subsequent derivations.

We may write the five parameters which define the parabola in vector form:

$$\mathbf{p} = (a, b, c, u, v)$$

Please note that the left hand side of (9.1) is the squared distance of point $(x,y) \in \mathbb{R}^2$ to the directrix, and the right hand side of (9.1) is the squared distance of point $(x,y) \in \mathbb{R}^2$ to the focus point. Therefore we can rewrite (9.1) as:

$$E_d(x,y) = E_f(x,y)$$

$$E_d(x,y) = \sqrt{\frac{(ax+by+c)^2}{a^2+b^2}} \tag{9.2}$$

$$E_f(x,y) = \sqrt{(x-u)^2+(y-v)^2} \tag{9.3}$$

where $E_d(x,y)$ is the distance of the test point (x,y) to the directrix, and $E_f(x,y)$ is the distance of the test point of the focus point. Furthermore, the plane is divided into two regions:

$$R_d = \left\{(x,y) \in \mathbb{R}^2 \mid E_d(x,y) \leq E_f(x,y)\right\} \tag{9.4}$$

$$R_f = \left\{(x,y) \in \mathbb{R}^2 \mid E_f(x,y) < E_d(x,y)\right\} \tag{9.5}$$

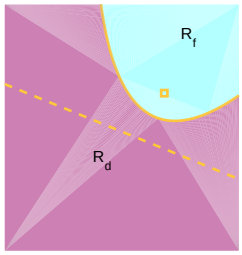


Figure 9.1: Depiction of the R_d and R_f regions. The directrix of the parabola is plotted with a dashed line, and the focus is drawn as a square.

where R_d contains the points which are closer to the directrix than to the focus point, or at the same distance, and R_f contains the points which are closer to the focus point than to the directrix (see Figure 9.1).

Now we may consider the minimization of the following cost function:

$$\mathcal{E}(\mathbf{p}) = \frac{1}{N} \sum_{i \in R_d} \left(E_f(x_i, y_i) - E_d(x_i, y_i) \right) + \frac{1}{N} \sum_{i \in R_f} \left(E_d(x_i, y_i) - E_f(x_i, y_i) \right) - \lambda E_d(u, v) \tag{9.6}$$

where N is the number of training points, the last term is introduced in order to avoid degenerate solutions with the focus on the directrix, and λ is an adjustable penalty parameter, $\lambda > 0$. Please note that the last term includes the distance from the focus point to the directrix, so that the degenerate solutions which must be avoided have $E_d(u, v) = 0$. Nondegenerate parabolas have $E_d(u, v) > 0$ which leads to $-\lambda E_d(u, v) < 0$. Degenerate parabolas with $E_d(u, v) = 0$ lead to $-\lambda E_d(u, v) = 0$ which is the maximum, i.e. worst since we are minimizing $\mathcal{E}$, possible value of the term $-\lambda E_d(u, v)$. In other words, the value of the term $-\lambda E_d(u, v)$ is always worse (higher) for degenerate parabolas than for nondegenerate parabolas.

The gradient of the error is computed as follows:

$$\frac{\partial \mathcal{E}}{\partial a} = \frac{1}{N} \sum_{i \in R_d} \frac{(ax_i + by_i + c)\left(ac + aby_i - b^2 x_i\right)}{\left(a^2 + b^2\right)^2 E_d(x_i, y_i)} - \frac{1}{N} \sum_{i \in R_f} \frac{(ax_i + by_i + c)\left(ac + aby_i - b^2 x_i\right)}{\left(a^2 + b^2\right)^2 E_d(x_i, y_i)}$$

$$+\lambda\frac{(au+bv+c)\left(ac+abv-b^2u\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{9.7}$$

$$\frac{\partial\mathcal{E}}{\partial b}=\frac{1}{N}\sum_{i\in R_d}\frac{(ax_i+by_i+c)\left(bc+abx_i-a^2y_i\right)}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$-\frac{1}{N}\sum_{i\in R_f}\frac{(ax_i+by_i+c)\left(bc+abx_i-a^2y_i\right)}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$+\lambda\frac{(au+bv+c)\left(bc+abu-a^2v\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{9.8}$$

$$\frac{\partial\mathcal{E}}{\partial c}=-\frac{1}{N}\sum_{i\in R_d}\frac{ax_i+by_i+c}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$+\frac{1}{N}\sum_{i\in R_f}\frac{ax_i+by_i+c}{\left(a^2+b^2\right)^2 E_d\left(x_i,y_i\right)}$$

$$-\lambda\frac{au+bv+c}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{9.9}$$

$$\frac{\partial\mathcal{E}}{\partial u}=\frac{1}{N}\sum_{i\in R_d}\frac{u-x_i}{E_f\left(x_i,y_i\right)}-\frac{1}{N}\sum_{i\in R_f}\frac{u-x_i}{E_f\left(x_i,y_i\right)}$$

$$-\lambda\frac{a\left(au+bv+c\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{9.10}$$

$$\frac{\partial\mathcal{E}}{\partial v}=\frac{1}{N}\sum_{i\in R_d}\frac{v-y_i}{E_f\left(x_i,y_i\right)}-\frac{1}{N}\sum_{i\in R_f}\frac{v-y_i}{E_f\left(x_i,y_i\right)}$$

$$-\lambda\frac{b\left(au+bv+c\right)}{\left(a^2+b^2\right)^2 E_d\left(u,v\right)} \tag{9.11}$$

It has been found in experiments that keeping the normal vector of the directrix normalized improves the numerical stability of the method, so that the obtained solutions are better. It is also advantageous to restart the search from a random state close to the initial estimation each T steps, because this allows the method to escape from local minima of the cost function $\mathcal{E}$.

The algorithm is:

1. Initialization. The focus point is initialized to the mean of the dataset:

$$(p_4,p_5)=\left(\frac{1}{N}\sum_{i=1}^{N}x_i,\frac{1}{N}\sum_{i=1}^{N}y_i\right)$$

The directrix is initialized to the straight line which best fits the dataset in the least squares sense:

$$(p_1, p_2, p_3) = \arg\min_{(a,b,c)} \sum_{i=1}^{N} (ax_i + by_i + c)^2$$

In order to keep a scale in the parameter space for the gradient descent, the directrix is normalized:

$$(\bar{p}_1, \bar{p}_2, \bar{p}_3) = \left(\frac{p_1}{\sqrt{p_1^2 + p_2^2}}, \frac{p_2}{\sqrt{p_1^2 + p_2^2}}, \frac{p_3}{\sqrt{p_1^2 + p_2^2}} \right)$$

Then a random perturbation drawn from a Gaussian distribution with zero mean and half the standard deviation of the dataset is added to the initial focus point. These are rough estimations, as the method will refine them later. The step counter is initialized to $t = 0$.

2. Determine the errors $E_d(x_i, y_i)$ and $E_f(x_i, y_i)$ for all training points (x_i, y_i) by (9.2) and (9.3).

3. Compute which training points belong to R_d and R_f with (9.4) and (9.5), where points on the boundary between both regions are assigned to one of them uniformly at random.

4. Find a new solution by gradient descent:

$$\mathbf{p}_{new} = \mathbf{p}_{old} - \eta \frac{\partial \mathcal{E}}{\partial \mathbf{p}}$$

where $\frac{\partial \mathcal{E}}{\partial \mathbf{p}}$ is given by Eqs. (9.7)-(9.11).

5. Normalize the directrix (as in Step 1) of the new solution in order to improve the numerical stability of the method.

6. Increase the step counter t. If the current solution has a lower value of the cost function (9.6), then store it as the best solution so far.

7. If the step counter t is an integer multiple of T, then restart the method by setting the current solution to a randomly perturbed version of the initial non perturbed solution. The random perturbation is carried out by adding a normally distributed random number to each of the components of the initial non perturbed solution of step 1. These random numbers are generated with zero mean and half the standard deviation of the dataset.

8. If convergence has been achieved or a maximum number of steps M has been reached, then return the stored best solution and halt. Otherwise, go to step 2.

The design of step 1 is aimed to obtain a valid solution, i.e. a non degenerate parabola. The random perturbation ensures that the perturbed solution is not degenerate. Input data which are aligned or almost aligned can lead to a degenerate initial parabola. Furthermore, the initial focus and the directrix are chosen to be close to the input data, so that the initial parabola has enough curvature, i.e. it is not too open. On the other hand, the normalization of step 5 ensures that the values of the directrix parameters a, b and c do not grow too large or become too small, which would cause loss of precision in the calculations and would ultimately destabilize the algorithm from the numerical point of view.

The convergence monitoring of step 8 is carried out by analyzing the updates of the current best solution which are done at step 6. Convergence is declared when the current best solution has not been updated for $2T$ steps.

It must be noted that our proposed algorithm might have some resemblances to the Expectation-Maximization (EM) algorithm. In particular, in step 3 of our algorithm it is estimated which side of the parabola each training sample is. This would be similar to the Expectation step, if we interpret that the "missing value" or "latent variable" would be a binary variable for each training sample which says which side of the parabola the sample is. Then in step 4 of our algorithm the five model parameters a, b, c, u, v are updated by assuming that the estimated values of such binary "latent variables" are correct, which is somehow similar to the Maximization step. However, our algorithm also has fundamental differences with respect to Expectation Maximization. First of all, no probabilistic model of the input data is defined or assumed. Therefore, there is no likelihood function or log-likelihood function to be optimized. This means that the EM algorithm cannot be applied to our parabola fitting approach. Secondly, unlike the Maximization step of the EM algorithm, our step 4 does not maximize the cost function given the estimated values of the binary "latent variables", but only moves the parameters in the direction of the gradient descent.

9.4 Experimental results

In this section several experiments are carried out to show the performance of the proposed method. Firstly, Subsection 9.4.1 briefs the tuning of the parameters of the algorithm. Then, some computational experiments for synthetic datasets are reported in Subsection 9.4.2. Simulations with and without outliers are presented, but also were analyzed datasets with Gaussian noise, degenerate parabolas and the effect of different initializations. Finally, real examples with data from natural images were used in Subsection 9.4.3.

Two of the competing approaches mentioned in Section 9.2 which have been tested are considered in (Harker et al., 2008), Direct Method and Cals Method. In addition to this, a third method has been used for the comparison, which was called as García Method (García-Escudero et al., 2017). For the experiments, fixed values $L = 400$ and $S = 40$ have been adopted following the advice of the authors of this method, and α with a value 0.1 higher than the outlier level, i.e. if we add 10% of outliers, we employ $\alpha = 0.2$. If no anomalous samples are present in the dataset, $\alpha = 0$ was used. Ours, Direct and Cals methods are implemented as MATLAB scripts (R2018a), with no use of the GPU. García method is implemented in R, but we linked it with MATLAB by the RMatlab package (Bengtsson et al., 2015) to perform all the comparisons under the MATLAB environment.

To measure the quality of the fit for the synthetic datasets, we compute the residual errors, consisting on the mean squared differences between the directrix and focus errors along the sample points:

$$Residual = \frac{1}{N}\sum_{i=1}^{N}\left(E_d\left(x_i, y_i\right) - E_f\left(x_i, y_i\right)\right)^2 \tag{9.12}$$

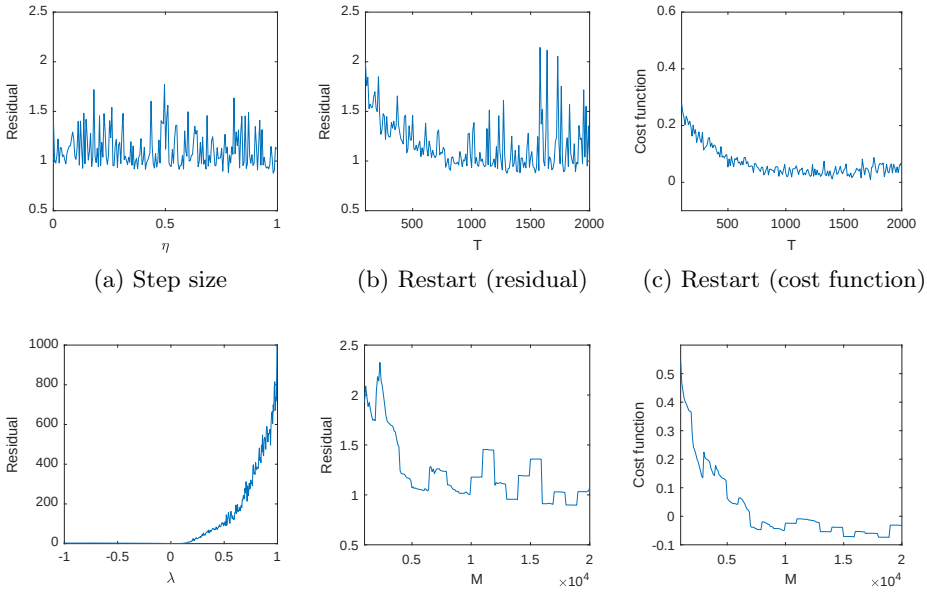
If spurious points (outliers) are added in the fitted dataset, these are not used to compute the Residual measure. This is done because the goal of a parabola fitting algorithm is to fit the inliers and not the outliers. Also, we obtained the runtime of all methods on a desktop personal computer with a 64-bit PC, with an Intel Core i7-4790, 3.6GHz CPU, 32 GB RAM and standard hardware.

9.4.1 Parameter selection

Our algorithm requires setting four parameters, the penalty parameter λ, the gradient descend step size η, the restart parameter T and the maximum number of steps M.

To determine the optimum values for the parameters, synthetic sets of points of a parabola were generated in the same manner as the first set of experiments detailed in Subsection 9.4.2. Then, we have run 100 executions of our algorithm and the Residual errors (9.12) were calculated for each execution varying the parameter of study and fixing the others. In addition to this, for the sake of completeness we have analyzed the cost function $\mathcal{E}$ (9.6) for the T and M parameters since there is no direct dependency on the parameters.

Figure 9.2 shows the results of the simulations. The average values of the measures along all the executions are depicted. In the case of the Step Size (Figure 9.2a), no significant differences can be appreciated between almost all possible values. The value 0.1 was chosen because is usually better to set a small value so that the gradient descent avoid abrupt jumps and the



(a) Step size (b) Restart (residual) (c) Restart (cost function)

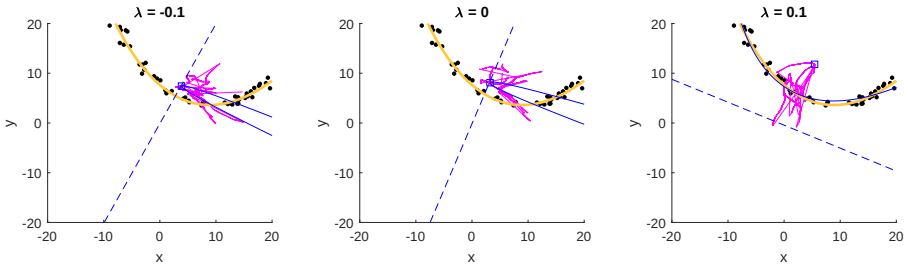
(d) Penalty parameter (e) Max. steps (residual) (f) Max. steps (cost function)

Figure 9.2: Study of the optimal parameters of our algorithm. Results of the residual errors and cost function for each parameter of our algorithm. Average results of 100 executions are displayed. The penalty parameter λ was varied from -1 to 1, and the gradient descend step size η from 0 to 1, both using a step of 0.005. The restart parameter T was varied from 100 to 2000, using a step of 10, and the maximum number of steps M from 1000 to 20000, using a step of 100. To analyze each parameter, the rest of them where fixed.

convergence is more reliable.

Regarding the restarting parameter T (Figures 9.2b-9.2c) and the maximum number of steps M (Figures 9.2e-9.2f), it can be seen that there is a moment from which the results stabilize for both parameters. This point was selected as our value for the tunable parameter.

With respect to the penalty parameter (Figure 9.2d), the testing of negative values of λ has been done for the sake of completeness. The condition $\lambda < 0$ means that we are rewarding degenerate solutions, which is expected to give bad results, but we still want to experimentally show that this is the case. And when $\lambda > 0.15$, the error increases exorbitantly. The penalty parameter is designed to avoid finding a degenerate parabola when the true input parabola is not degenerate. To show the effect of λ we have carried out an experiment where we create an input dataset corresponding to a regular, i.e. non degenerate, parabola, and we have run our algorithm for $\lambda = -0.1$,



(a) Negative value (b) Zero value (c) Positive value

Figure 9.3: Outcomes of our algorithm for a non degenerate parabola when penalty parameter is $\lambda = -0.1$, $\lambda = 0$ and $\lambda = 0.1$, from left to right. The rest of the parameters where fixed.

Table 9.1: Parameter selection of our method.

Parameter	λ	η	T	M
Value	0.1	0.1	1000	10000

$\lambda = 0$ and $\lambda = 0.1$. Figure 9.3 shows that the obtained solutions are degenerated for the first two cases, as it was expected, and the proper solution is reached with the positive value.

The previous discussion is specific to a particular dataset. However, the resulting parameter selection has worked for all the other tested datasets, so the subsequent experiments validate the parameter values chosen in Subsection 9.4.1. All the selected parameter values for our method are summarized in Table 9.1.

9.4.2 Synthetic data

Our first set of experiments deals with artificially generated data. For each experiment, the focus and the directrix of a parabola are chosen at random:

$$u, v \sim U(0, 20) \tag{9.13}$$

$$a, b, c \sim U(0, 30) \tag{9.14}$$

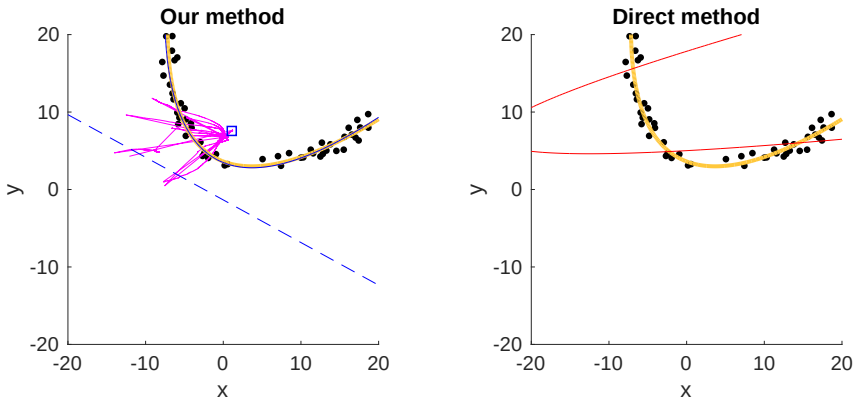
where $U(\alpha, \beta)$ stands for the uniform distribution over the open interval of real numbers (α, β). Then 100,000 candidate points are uniformly generated on the square $[-20, 20] \times (-20, 20)$. Then, we select the points (x_i, y_i) whose which satisfy that:

$$(E_d(x_i, y_i) - E_f(x_i, y_i))^2 < 2 \tag{9.15}$$

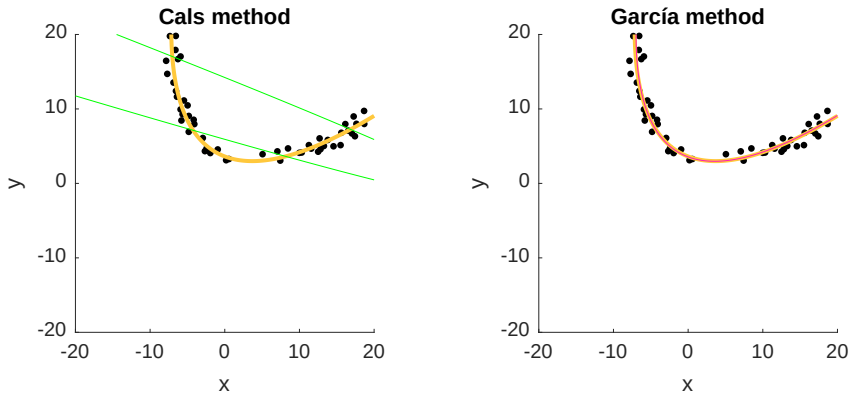
Note that (9.15) means that the input samples are close to the true parabola, which is given by:

$$\left(E_d\left(x_i, y_i\right) - E_f\left(x_i, y_i\right)\right)^2 = 0 \tag{9.16}$$

but they are not exactly on the true curve, i.e. some noise is present. The value 2 was manually chosen so that the noise was significant but not so large that the parabolic shape was lost. Finally, from this candidates are randomly selected the input datasets with size $N = 50$.



(a) Our method (b) Direct method

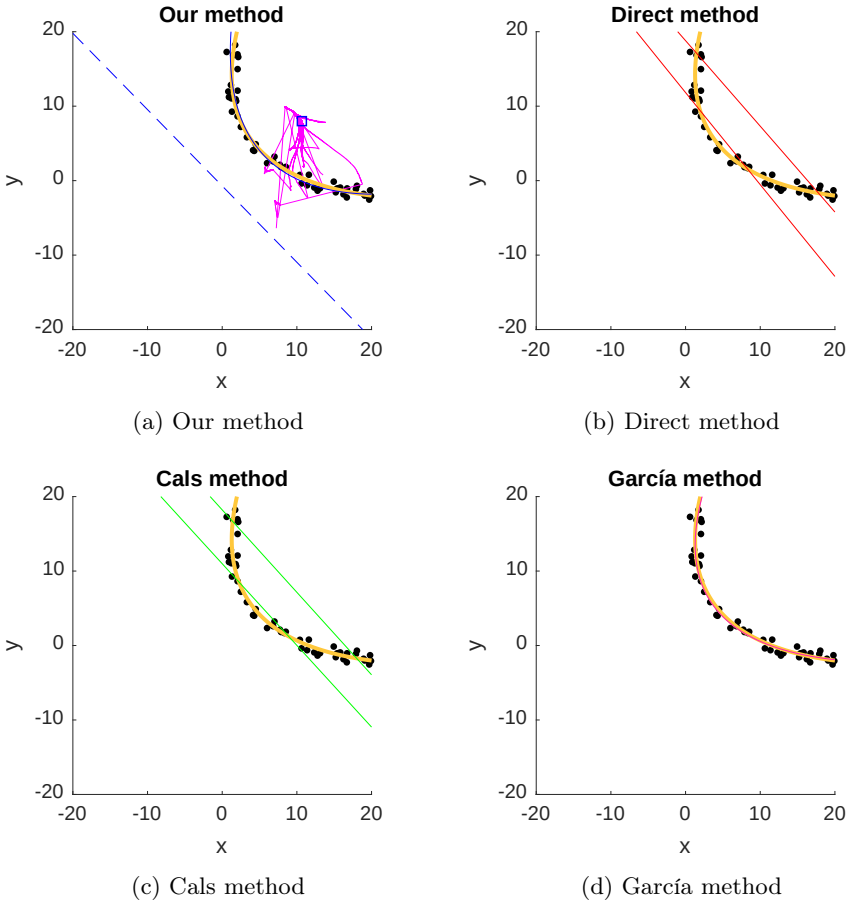


(c) Cals method (d) García method

Figure 9.4: Solutions for the first synthetic dataset. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

If a percentage of outliers is added in the experiments, then the number of those is calculated as a fraction of N. That is, if we add $p_{out}\%$ of outlying

points, then the final number of samples is $N + \frac{p_{out} \cdot N}{100}$. These extra points are randomly selected from those candidate points that do not satisfy (9.15), i.e., they are points where $(E_d(x_i, y_i) - E_f(x_i, y_i))^2 > 2$.



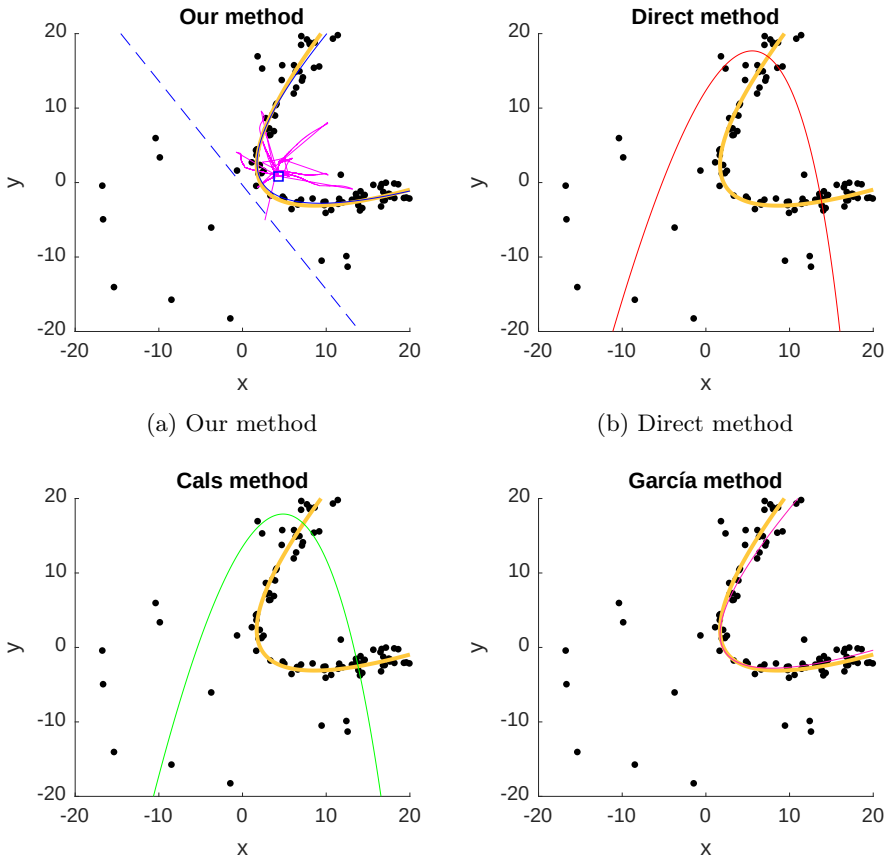
(a) Our method (b) Direct method

(c) Cals method (d) García method

Figure 9.5: Solutions for the second synthetic dataset. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

Next we present the results of the competing methods for some cases. First example is shown in Figure 9.4. A synthetic dataset was generated in absence of anomalous samples. Comparing all the fits, both ours and García methods yield the best results. As seen in Figure 9.4a, the trace of the estimated foci sometimes goes far away from the true solution. This is due to the existence of many spurious solutions which have good values of the cost function $\mathcal{E}$. And this is why Direct and Cals methods return incorrect estimations of the parabola, falling in a local minima. However, this issue

has been addressed in our proposed algorithm by introducing restarts from randomly perturbed versions of the initial non perturbed solution (step 7 of the algorithm) at regular intervals. These restarts are seen in Figure 9.4a as line segments which come back to the vicinity of the true focus.



(a) Our method

(b) Direct method

(c) Cals method

(d) García method

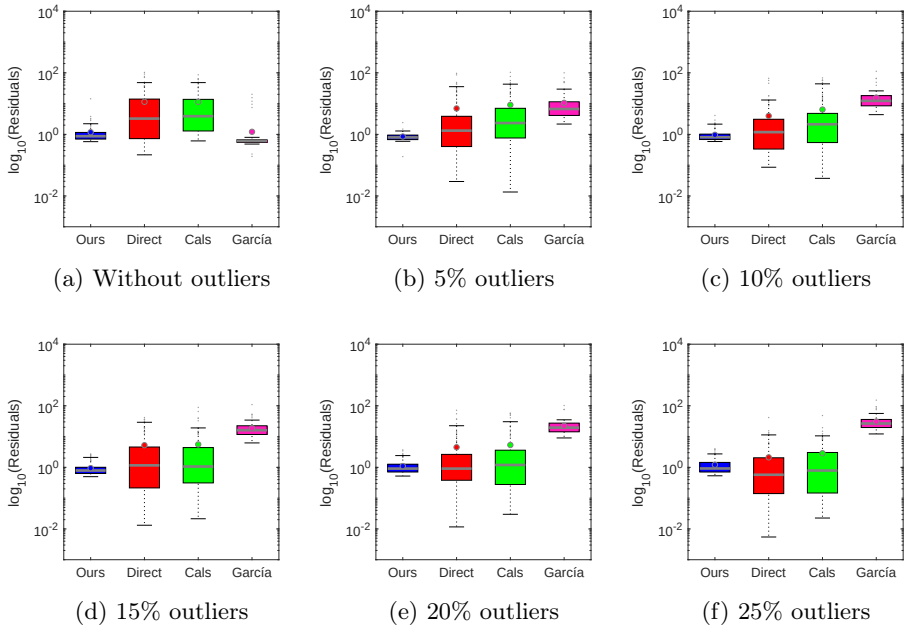
Figure 9.6: Solutions for the third synthetic dataset. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

In Figure 9.5 a second example dataset is presented. This dataset features a higher occlusion level, in the sense that the points corresponds to a small arc of the parabola, becoming difficult to determine which type of conic they are part from, so the fit is exposed to an incorrect fitting. As it is shown Direct and Cals methods yield similar results, neither of them converge to a good solution. The first and second synthetic datasets are similar, but it is relevant to depict their results because they are largely different for the

unstable methods (Direct and Cals).

The last example dataset is shown in Figure 9.6, where a 20% of outliers is added to the training data. The results obtained by our method and García method are quite acceptable, while Direct and Cals methods show a clearly wrong outcome. It seems that in most of the examples, our approach have a similar or even better fit to the point cloud than García algorithm.

In general terms, it can be said that our method exhibits a higher resilience to noise than its competitors for synthetic data. To further confirm this, a set of 100 executions has carried out and the residual errors are shown in Figure 9.7. Simulations without outliers and with 5%, 10%, 15%, 20% and 25% of added outliers have been done. The outliers are not considered in the computation of the residuals. Boxplots were used to display the results. We have used the Residual measure defined in (9.12) to compare the results. As it can be seen, in almost all cases our method yields the best results, as the mean squared differences between the errors $E_d(x_i, y_i)$ and $E_f(x_i, y_i)$ along the training points are very close to zero, that is, the samples are very close to the fitted parabola. In the case of median values, the different approaches are more competitive but they have a higher deviations, as it can be seen in the higher size of the box.



(a) Without outliers (b) 5% outliers (c) 10% outliers

(d) 15% outliers (e) 20% outliers (f) 25% outliers

Figure 9.7: Comparative of the four methods along 100 executions with synthetic data. Grey circles and lines represent the mean and median value over all the simulations, respectively. Displayed whiskers are in the range of 5-95%. Results shown in a logarithmic scale.

Table 9.2: Mean and standard deviation of the running time (in seconds) along the 100 executions for different noise levels.

	Ours	Direct	Cals	García
0%	0.333 (0.036)	0.001 (0.005)	0.003 (0.009)	79.554 (1.818)
5%	0.334 (0.012)	0.001 (0.003)	0.003 (0.006)	84.087 (1.651)
10%	0.334 (0.006)	0.001 (0.000)	0.002 (0.000)	86.313 (1.390)
15%	0.3348 (0.006)	0.001 (0.000)	0.002 (0.000)	90.916 (1.584)
20%	0.342 (0.017)	0.001 (0.000)	0.002 (0.000)	93.654 (1.594)
25%	0.347 (0.040)	0.001 (0.000)	0.002 (0.000)	97.469 (1.972)

The results are very similar when we have no outliers (Figure 9.7a). Clearly, Direct and Cals methods yield inconsistent results, while García approach obtains good performance. Figures 9.7c and 9.7b depict the results for adding 5% and 10% of anomalous points to the initial dataset, respectively, which implies the presence of few points that do no belong to the initial selection of points near to the parabola since Eq. (9.15). For this configuration, the deviation errors are quite similar to that one without outliers with the exception of García method, which suffers big mismatches as the number of outlying points increases. It is remarkable that the residuals, which are computed taking only into account the inliers, preserve the goodness of fit only in our case.

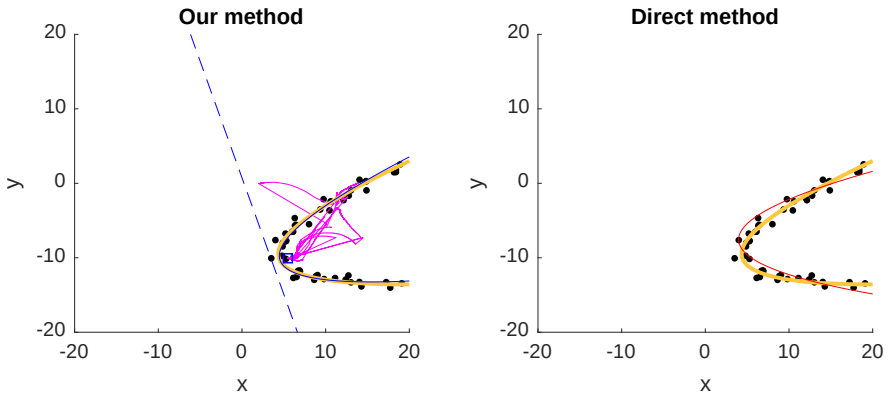
The statistics when 15%, 20% and 25% of outliers is added to the training points are shown in Figures 9.7d, 9.7e and 9.7f. The outcomes of these three configurations are very similar. Although our method has the best results, Direct and Cals methods reduce the difference in terms of median values. However, these competing methods generate solutions with to much deviation. The mean and standard deviation of our proposal still remain clearly the best. There could be cases where Direct and Cals methods yield small values of the residuals, even if the number of outliers increases. That is because the residuals are computed setting aside outliers.

Table 9.2 summarizes the computational time required for the execution of the four methods. As we can see, our method is the only one that has an equilibrium between performance and computational efficiency. The mean running time employed by our approach is always less than 0.35 seconds. We can also see that Direct and Cals methods fit their parabolas faster than ours, but with the inconvenience of inconsistent results. García method has demonstrated good accuracy in some cases, but the required time to compute the parabola is excessive, it is around 90 sec., which is one hundred times slower than the other methods.

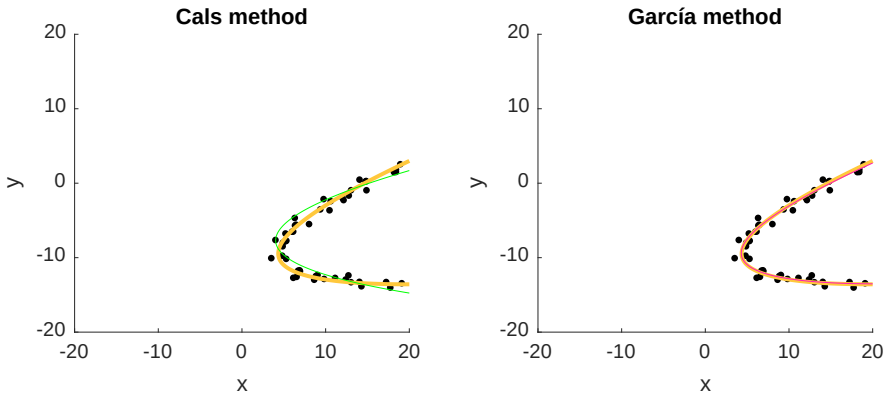
9.4.2.1 Gaussian distribution

A second set of experiments with synthetic data was carried out. Starting from the unit parabola with equation $y = x^2$, any other parabola can be inferred from this one using an affine transformation of the Euclidean plane in the form $\mathbf{x} \to \mathbf{f_0} + \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is a regular matrix with column vectors $\mathbf{f_1}, \mathbf{f_2}$, and $\mathbf{f_0}$ is an arbitrary vector. Thus, the general parametric equation of a parabola is defined as

$$\mathbf{x} = \mathbf{f_0} + \mathbf{f_1}t + \mathbf{f_2}t^2,\ t \in \mathbb{R} \tag{9.17}$$



(a) Our method (b) Direct method



(c) Cals method (d) García method

Figure 9.8: Solutions for a synthetic dataset with Gaussian noise distribution. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

As before, 100,000 candidate points on the square $[-20, 20] \times (-20, 20)$

are generated following (9.17) with $\mathbf{A} = \mathbf{R}\Lambda$, where Λ is a diagonal matrix which is multiplied by a 2D rotation matrix $\mathbf{R}$. Then, Gaussian noise with standard deviation 0.5 was added to the points of the parabola. Finally, a subset with size $N = 50$ from this candidates are randomly selected.

In Figure 9.8 we present an example of an execution with this kind of dataset. Both Ours and García methods yields the best results in comparison with the true parabola. Direct and Cals methods also show a good fit, but not as good as expected since the LS approaches assume normally distributed residuals.

For an adecuate evaluation, we carried out a set of 100 execution with different parabolas generates as explaned before. The results are summarized in Figure 9.9. Our approach yields great results, specially in terms of small deviations. If we compare the outcomes obtained from the previous experiments (Figure 9.7), the Direct and Cals methods show lower errors but similar deviations, and García yields higher errors.

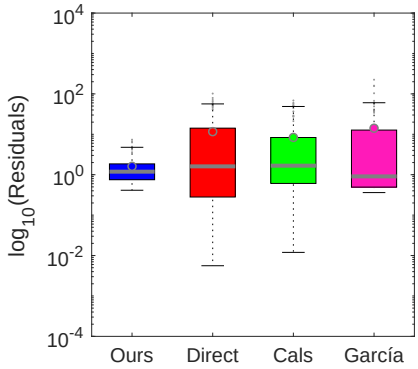
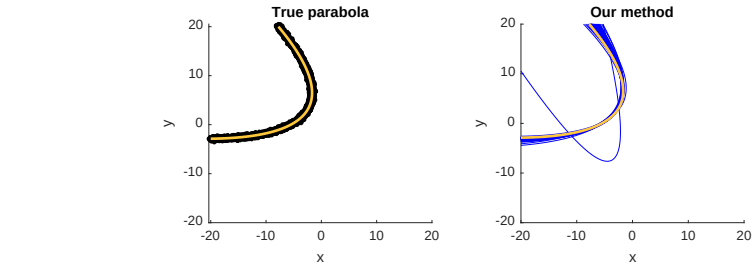


Figure 9.9: Comparative of the four methods along 100 executions with synthetic data and Gaussian noise distribution. Results shown in a logarithmic scale.

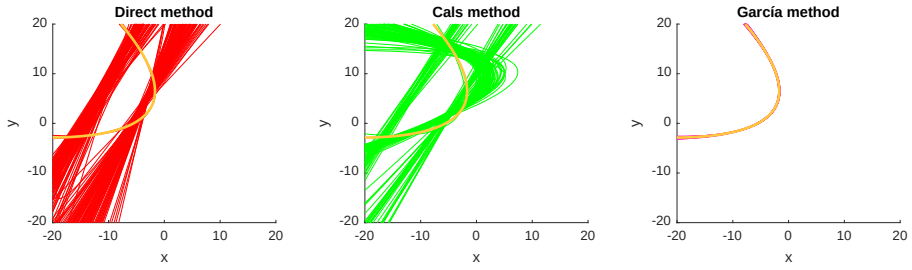
9.4.2.2 Variability of the fitting

In the next experiment, we see the differences in estimates for different initializations in order to study the variability of the algorithms. Thus, we construct a parabola as is explained in 9.4.2.1, and Gaussian noise with standard deviation 0.2 was added to the candidate points defined on the parabola. The generated cloud points is depicted in Figure 9.10a. Then, we carried out a total of 100 executions with different sample points with size $N = 50$, selected randomly from the set of candidate points.

In Figure 9.10 all the solutions for all the initializations are represented, so we can have a qualitative assessment of the four algorithms. Clearly,



(a) True parabola (b) Our method (0.3495 s.)



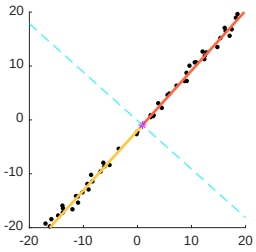
(c) Direct method (0.0011 s.) (d) Cals method (0.0025 s.) (e) García method (81.7085 s.)

Figure 9.10: Solutions for all the different initializations. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve. Each caption contains the mean execution time needed to obtain the fitted parabola.

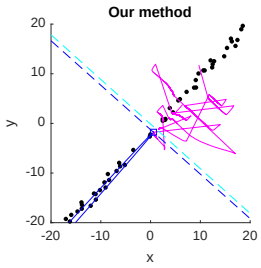
Direct and Cals methods fails too many times in the predictions, which is not a good symptom in terms of robustness. The most stable method is García. However, it is important to remark that it needs a lot of time to process the information and generate the solution, which could be an impediment for some real time applications. Our approach also achieves a good performance, since it only fails once, and its computing time is quite acceptable.

9.4.2.3 Degenerated parabola

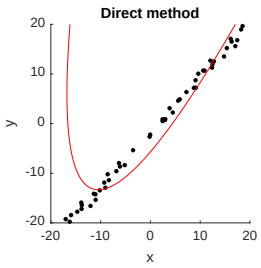
In this subsection we analyze the behavior for all methods in the case of a degenerated parabola. To generate it, a random directrix was computed following (9.14) and a point in this line has been chosen as the focus. Thus, two symmetric branches are the two solutions of this configuration, as is shown in Figure 9.11a in colors yellow and brown. These degenerate parabolas are orthogonal to the directrix, which is shown in a dotted light blue line. To generate the dataset, we followed the methodology explained at the beginning of the experimental section but varying the error margin (9.15) to a



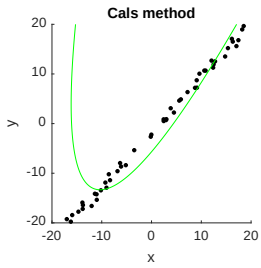
(a) True degenerated parabolas



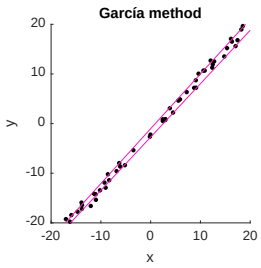
(b) Our method



(c) Direct method



(d) Cals method



(e) García method

Figure 9.11: Solutions for a synthetic dataset with Gaussian noise distribution. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. The true parabola is plotted as a thick solid curve, while the fitted parabola is shown as a narrow solid curve.

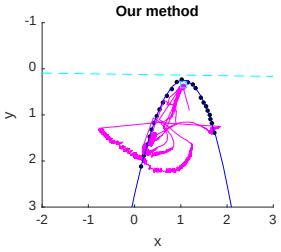
value very close to zero. In this example we used an error of 0.001. Residuals were also computed in order to compare the methods from a quantitative point of view.

The best approximation to the real degenerated parabola is achieved by our proposal, which is the only one that finds one of the two solutions. It is not a straight line, but the obtained parabola is the flattest one. The Residual error is 0.000502, which is very close to zero. In addition, the estimated directrix is almost the same as the true one. Both Direct and Cals methods fail in the prediction, with Residuals 3.7190 and 3.4084 resp., and García approach estimates a flattened parabola but the focus is wrongly estimated to lie at one extreme of the data domain. However, the Residual is the best one, 0.0000207.

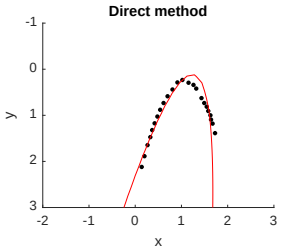
9.4.3 Real data

Next some real dataset examples are presented such as a fountain, a denture sample, a bridge and a reflector. We have overlayed the data points on each

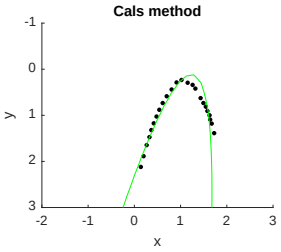
(a) Fountain example: real image



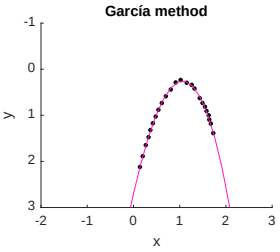
(b) Our method (0.4042 s.)



(c) Direct method (0.0390s.)



(d) Cals method (0.0458 s.)



(e) García method (40.2343 s.)

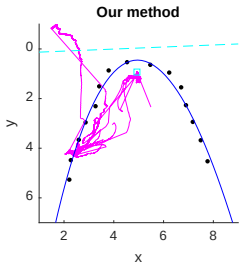
Figure 9.12: Solutions for the fountain dataset. Points were manually selected from one of the water streams. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format.

image for the sake of clarity.

The first two examples, the fountain and the denture sample, are based on manually selected points on the parabolic curves defined by them. As shown in Figure 9.12 our results are better than the competing methods. As known, the trajectory of the water drops or any other bodies under the sole influence of the gravitational field of the Earth is a parabola. Although all the algorithms return a fitted parabola very similar that one described by the fountain, Direct and Cals fail in the orientation. Both Ours and García generate good quality solutions but with very different CPU times.

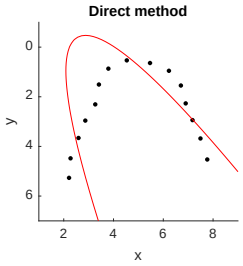
In the case of the denture dataset (see Figure 9.13), the centers of the teeth should describe a parabola for an ideal denture. Having this in mind, this example can be considered as a dataset with a small level of noise. It is shown that both García and our methods do not fail and have a better performance than the other two methods.

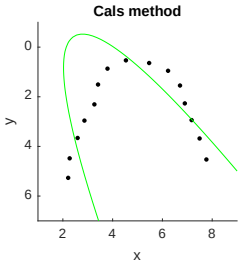
Next, we used the image intensities to extract the bridge sample dataset

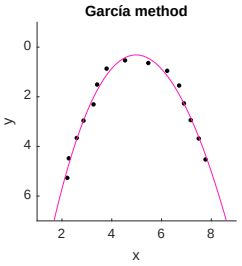


(a) Denture example: real image

(b) Our method (0.4271 s.)





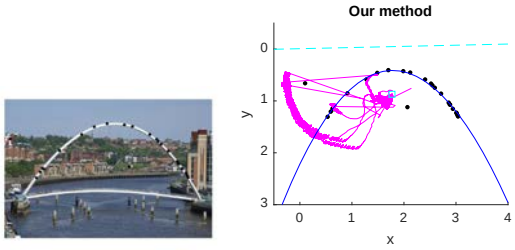


(c) Direct method (0.0396 s.) (d) Cals method (0.0471 s.) (e) García method (42.8631 s.)

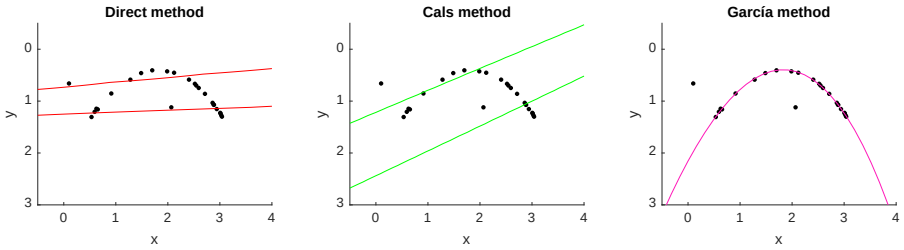
Figure 9.13: Solutions for denture real data. Points were manually selected from the center of the teeth. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format.

(see Figure 9.14a). A white color threshold was applied followed by the application of a region of interest (ROI) mask to remove all pixels that are below the bottom part of the bridge arc, in order to remove the crossing pathway. Then, a random selection of 25 points from the segmented image were carried out. As it can be seen in Figure 9.14a, there are two spurious points in the final data selection, i.e., 10% of outliers approx. Figure 9.14 demonstrate that our method is robust against anomalous data, achieving a great fit compared to Cals and Direct methods, which are unstable when the dataset does not describe a large arc of the parabola. García algorithm works very well but only if the trimming level is provided. Otherwise, i.e., $\alpha = 0$ the method fails in the fit. This parameter is hard to determine in real applications.

An industrial real data example is used in Figure 9.15. In this case, the dataset describes a parabola with a radius of curvature a little higher than the previous examples. The Canny edge detection method has been employed to extract edge points on the parabolic curves from the image,



(a) Bridge example: real image (b) Our method (0.4582 s.)



(c) Direct method (0.0403 s.) (d) Cals method (0.0507 s.) (e) García method (43.5763 s.)

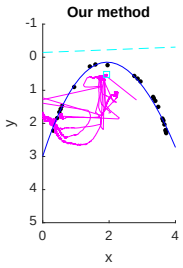
Figure 9.14: Solutions for the real data about a bridge. 25 points were randomly selected using a color threshold for pixels with values greater that 214 (in a scale [0-255]) for the three RGB channels, followed by a ROI with $0 < X < 215$ in the first dimension. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to concise with the image visualization format. We used $\alpha = 0.2$ as an input for García method.

and then the training samples have been randomly drawn from the set of edge points.

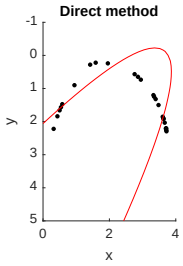
The dataset shown in Figure 9.15a exhibits three anomalous points near the vertex that clearly are not part of the parabolic shape. As seen, all the competing methods present worse results than ours (see Figure 9.15). Direct and Cals fail in the fit and García method generates a good fit, but still it is not as precise as ours and also the trimming level needed to be provided. García only works well if the true percentage of outliers is supplied, which would not be easy to get in a real application, while our method does not need that input.

Overall our proposal achieves a good fit to the above presented real datasets, and employing very little time to compute the fitted parabola. Therefore, its robustness against noise is further validated. It is also re-

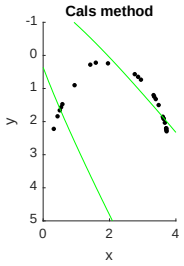
(a) Reflector example: real image



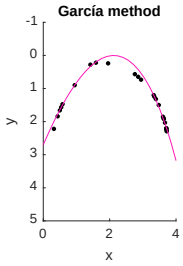
(b) Our method (0.4121 s.)



(c) Direct method (0.0312 s.)



(d) Cals method (0.0393 s.)



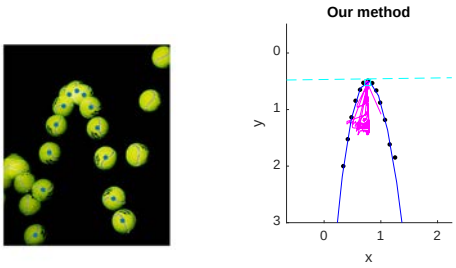
(e) García method (39.6732 s.)

Figure 9.15: Solutions for the real data about a reflector. 25 points were randomly selected from the output of the Canny edge detection function of MATLAB with threshold=0.95, followed by a ROI with $0 < X < 375$ in the first dimension. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to concide with the image visualization format. We used $\alpha = 0.2$ as an input for García method.

markable that the method do not need to be provided by input parameters based on the employed dataset.

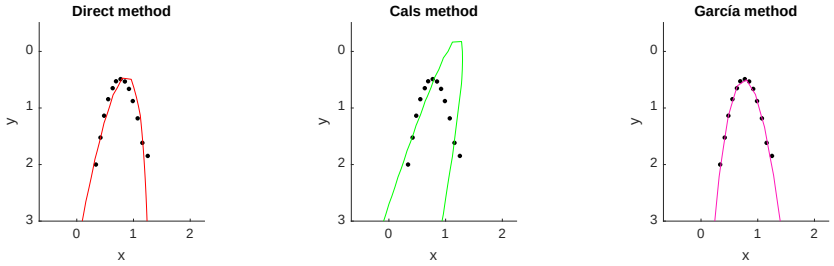
The following examples are based on stroboscopic images of spherical objects that describe a parabolic shot. In this case, the point were automatically extracted using the Circle Hough Transform, which is a feature extraction technique used to find imperfect instances of objects by a voting procedure. It has a great performance to detect object that have circular or elliptical shapes. MATLAB *imfindcircles* function was used to detect the objects in the image and determine

Figures 9.16 and 9.17 show two examples of the execution of the algorithms with stroboscopic images. In Figure 9.16a we can see the movement of a tennis ball throwing and describing three parabolas. We select the middle one and extracted the points as specified in the figure caption. We



(a) Tennis ball example: stroboscopic image

(b) Our method (0.3988 s.)



(c) Direct method (0.0304 s.)

(d) Cals method (0.0447 s.)

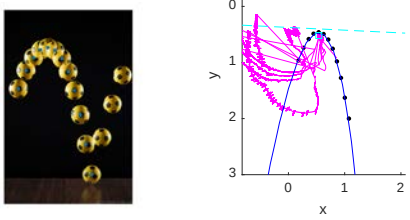
(e) García method (38.4468 s.)

Figure 9.16: Solutions for tennis ball stroboscopic image. Points were selected using the Circle Hough Transform through MATLAB with Radius Range = [15 45] and Sensitivity = 0.9 as input parameters. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format. CPU time is displayed in subcaptions.

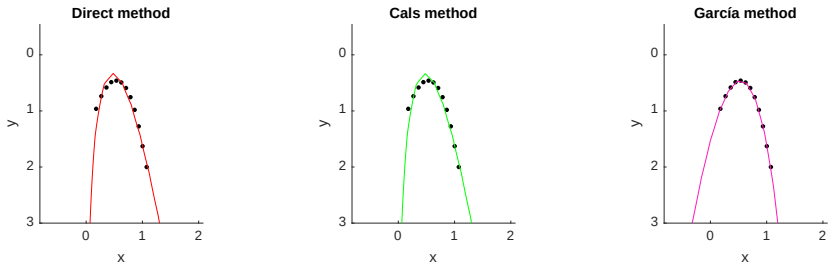
can observe that the best fits are achieved again by our proposal and García method, but this one employs too much time although the number of sample points is very small.

In Figure 9.17a a soccer ball is pictured. The first parabola is selected, which has an occluded part. However, this does not affect the estimations of the four methods, but Direct and Cals methods still have a bit worst performance.

If we compare all the processing times for all the experiments that we have carried out with real examples, we can see that the fastest ones are always Direct and Cals, with around 0.05 seconds. The third position is occupied by our algorithm, with a mean value of 0.42 seconds, but with the difference that the fitting performance is clearly better. At last, García method is very robust but needs more that 40 seconds to obtain the result.



(a) Soccer ball example: stroboscopic image (b) Our method (0.4122 s.)



(c) Direct method (0.0318 s.) (d) Cals method (0.0666 s.) (e) García method (36.4591 s.)

Figure 9.17: Solutions for soccer ball stroboscopic image. Points were selected using the Circle Hough Transform through MATLAB with Radius Range = [15 45] and Sensitivity = 0.85 as input parameters. In magenta the trace of the foci of our method is represented, and the blue square and the dotted line are the best foci and directrix achieved. Fitting results are plotted in a reverse Y scale to coincide with the image visualization format. CPU time is displayed in subcaptions.

9.5 Conclusions

This research presents a parabola fitting method based on minimization of the parabola geometric function to a set of points by minimizing the mean absolute deviation from the level set, which defines the parabola feature. The algorithm comprises of two stages: 1) determining the closed distance between data points to the directrix and to the focus point, and 2) solving an absolute geometric cost function for parameter estimates with a normalized directrix vector that improves the stability of the method. Furthermore, the search is restarted each certain number of steps in order to escape from local minima. This way the proposed algorithm converges to a finite solution irrespective of the initial parameters.

Experimental results demonstrate that our algorithm is robust since the

solutions have a low sensitivity to noise data. The robustness of the proposed algorithm is due to the usage of absolute rather than squared errors, since the sum of absolute errors is more robust than the sum of squared errors. State of the art alternative methods have been tested, which are either much slower than ours, or clearly worse in their fitting accuracy. Therefore, it can be concluded that a fast and accurate approach for parabola fitting has been proposed, which can be used for datasets with significant amounts of noise and occlusion. This facilitates its application to real problems with low quality input data.

Directions for future work include further improving the accuracy of the estimation of the parabola parameters, while keeping the computational complexity low. As seen in the experiments, García method is quite robust, and it sometimes finds better estimations of the parabolas, but at the expense of being one or two orders of magnitude slower than our approach. Hence there is still room for enhancing our approach. This might be done by hybridizing the algorithm presented here with some optimization framework such as genetic algorithms, tabu search or particle swarm optimization. Another possible strategy is the combination of several high quality solutions, in the spirit of ensemble learning.

Chapter 10

Ellipse fitting by spatial averaging of random ensembles

Random numbers should not be generated
with a method chosen at random

Donald Knuth

Abstract: Earlier ellipse fitting methods often consider the algebraic and geometric forms of the ellipse. The work presented here makes use of an ensemble to provide better results. The method proposes a new ellipse parametrization based on the coordinates of both foci, and the distance between them and each point of the ellipse where the Euclidean norm is applied. Besides, a certain number of subsets are uniformly drawn without replacement from the overall training set which allows estimating the center of the distribution robustly by employing the ℓ_1 median of each estimated focus. An additional post-processing stage is proposed to filter out the effect of bad fits. In order to evaluate the performance of this method, four different error measures were considered. Results show that our proposal outperforms all its competitors, especially when higher levels of outliers are presented. Several synthetic and real data tests were developed and confirmed such finding.

10.1 Introduction

Curve fitting is the process of specifying a model based on a particular curve, such as a circle, ellipse or parabola that provides a good fit to a set of points (Muñoz-Pérez et al., 2014; Nurunnabi et al., 2018; Gontar et al., 2018; López-Rubio et al., 2018). Fitting has become an important research topic in the

last years in many areas of science and technology. Also, this process is the first step for a large number of applications, for example, engineering applications that may include trend analysis, extrapolation or interpolation between data points.

In this work, a new method to fit an ellipse to a set of input points is presented. Our approach introduces ensemble methods into the curve fitting research field, which has not been previously proposed in the literature. This methodology differs from our previous works for other conic curves (López-Rubio et al., 2018).

Our proposal is motivated by the observation that several alternative solutions can be obtained from the same base algorithm by slightly changing the set of input points. Therefore, an enhanced fit can be computed by robustly combining those different solutions, so that grossly erroneous fits have a small effect on the final solution. In order to attain such a goal, a parametrization of the ellipse is proposed which is amenable to the robust averaging of the combined solutions. The median is employed to carry out such averaging. The rationale behind this is that the median is resilient to outliers, so that some completely wrong solutions do not ruin the overall estimation. The research gap to be addressed here is how accurate ensemble averaging of ellipse fits can be, as compared to the individual fits by classic algorithms which do not consider ensembles. Given the previous successes of ensemble methods when applied to other research fields, and the lack of ensemble proposals for ellipse fitting, the significance of our investigation is guaranteed. The validity of our proposal is assessed by quantitative and qualitative studies of its performance, as compared to state-of-art ellipse fitting algorithms.

The structure of the chapter is: Section 10.2 presents state-of-art techniques related to ellipse fitting and ensembles. Section 10.3 describes the theory and the proposed algorithm and after, the results and discussion of the experiments are presented in Section 10.4. At the end, Section 10.5 reports the conclusions and possible further works.

10.2 Background

10.2.1 Fitting ellipse in literature

Ten fundamental methods used in this research are presented below. First, Taubin's method (Taubin, 1991) is a nonlinear least squares problem that consists in a non-iterative curve fitting method. It is based on implicit representation to a dataset minimizing the approach mean square distance where statistical properties of noise are not considered. Second, Szpak's method (Szpak et al., 2015) introduces an ellipse estimation procedure which endorses an optimization of the Sampson distance reached for a particular alterna-

tive of the Levenber-Marquardt algorithm. Third, Fitzgibbon's method (Yu et al., 2012; Fitzgibbon et al., 1996) is an algebraic method that minimizes the algebraic distance. This method fits an ellipse including ellipticity constraint into the normalization factor. Fourth, Muñoz's method (Muñoz-Pérez et al., 2014) is a robust multicriteria algorithm based on the mean absolute error. Fifth, Halir & Flusser's method (Halir and Flusser, 1998) presents an approach based on a least squares minimization. This method is non-iterative and stable from a numerical point of view. It is simple and efficient even in the presence of a large amount of noisy data, hence recommended for an initial robust ellipse estimation. Sixth, Rosin's method (Rosin, 1999) is an accurate and robust method based on least median of squares. The median of intrinsic parameters of the supposed ellipse is used to reckon the geometric parameters of the ellipse. Seventh, Prasad's method (Prasad et al., 2013) proposes an efficient least squares ellipse fitting method without any constrained optimization that is stable under high levels of noise. Eighth, Köning's method (Köning et al., 2014) proposes a geometric fitting to an ellipse minimizing measured and fitted signal values' distance. This method supplies model parameters' best linear unbiased estimators and includes the statistical uncertainties. Ninth, Liang's method (Liang et al., 2013) proposes a linear combination of a subset of few data points to construct the ellipse parameters. Also, it employs the absolute residuals to reduce the contribution of extreme data points and binds the position error of data points to be robust against the worst case. Finally, Sobhani et al. (Sobhani et al., 2017) propose a method based on algebraic distance minimization which uses the robust Huber's function. Their algorithm detects inliers that can determine the ellipse parameters.

10.2.2 Ensemble methods

In literature, some methods to solve problems by ensembles of solutions have been described. However, most of them are applied to classification issues (Amasyali, 2019; Pham and Olafsson, 2019). The accuracy of the ensemble is linked to the performance of each of the base algorithms used as well as the variety of their results. Choosing between more exact but less varied based learners and vice versa can actually be a rightful definition of distinct ensemble methods. In general, better outcomes are obtained since the process of blending and averaging makes the variance of the ensemble model decrease. The output of an ensemble depend on the individual achievement and the independence of the effect of each learner, i.e. high diversity and low error (Breiman, 2001). A common procedure to generate diversity is the use of different data subsets to train each base algorithm of the ensemble. In order to bring about different training data sets, some approaches such as random methods (Ho, 1998; Breiman, 2001; Geurts et al., 2006; Durrant and Kabán, 2015) and rotating (rotation forest) (Rodríguez et al., 2006) are employed.

Regarding the most well-known and useful ensemble methods are bagging, AdaBoost and LogitBoost (Bui et al., 2019). Bagging (bootstrap aggregating) ensemble method was introduced in the mid-'90s by Breiman (1996). He uses a bootstrap sampling technique, guaranteeing improvements in terms of accuracy as it reduces the classification error variance. The second procedure is the AdaBoost method, the most extensively used boosting algorithm published by Freund and Schapire (1996). It entails the implementation of an adaptive re-sampling method with improved predictive behavior since it reduces both bias and variance (Schapire, 2003). The third kind is the LogitBoost, a boosting method presented by Friedman et al. (2000), the idea of which is the bias and variance reduction (Song et al., 2011). LogitBoost process is considered as an expansion of the AdaBoost method (Tien Bui et al., 2016; Kadavi et al., 2018). Lopes (2019) proposes a bootstrap process to estimate this variance for bagging, random forests and related algorithms in the context of classification.

Notwithstanding the above, Rosin (1993b) uses a method that stores five-point ellipse fits as an ensemble and then determines the fit using medians. The Rosin's method advantages comprehend the high amount of outliers permitted without requiring any random parameters and its reliability owing to the lack of problems with convergence as in others least-squares based methods. Rosin uses the natural parametrization of the ellipse, based on the center coordinates, the length of the major and minor axes and the orientation. This means that with every five points subset an ellipse is fitted. When the outcome curve does not match an ellipse, the five-point subset is discarded. Taking all combinations of five points leads to an undesirable situation due to the great number of ellipses generated even if a small number of points are present. To accelerate this process, only a random sample of the original set of points that have not been employed yet is regarded. By doing this, each point of the initial data set is ensured to be included in one of the 5-tuple fits.

10.3 The SAREfit method

In this section our ellipse fitting method is detailed. Subsection 10.3.1 introduces our proposal, while Subsection 10.3.2 is devoted to the study of its robustness against outliers.

10.3.1 The algorithm

Let us consider an ellipse fitting algorithm φ that, given an input set of points T of size N, $T = \left\{\mathbf{x}_i \in \mathbb{R}^2 : i \in \{1, ..., N\}\right\}$ produces an output ellipse fit:

$$\begin{aligned} \varphi :\ & \mathbb{R}^{2N} \to \mathbb{R}^5 \\ & \varphi\left(T\right) = \boldsymbol{\psi} \end{aligned} \tag{10.1}$$

where $\boldsymbol{\psi} \in \mathbb{R}^5$ is a suitable parametric representation of the fitted ellipse which is obtained as the output of the algorithm. Now, if the overall training set $\mathcal{S}$ contains more than N samples, we may consider the 5-dimensional probability distribution of $\boldsymbol{\psi}$ when T is randomly drawn from $\mathcal{S}$ with replacement. Such multivariate distribution on $\mathbb{R}^5$ is expected to be unimodal, where the center of the distribution is close to the true ellipse that is to be estimated. In other words, the samples of $\boldsymbol{\psi}$ are expected to cluster around the point in $\mathbb{R}^5$ which represents the true ellipse under the considered ellipse parametrization. The less biased the algorithm, the closer the center of the distribution to the true ellipse. Depending on the proportion of outliers in $\mathcal{S}$, some of the realizations of $\boldsymbol{\psi}$ are noisy, i.e. some ellipse fits are bad estimates of the true ellipse. In order to overcome this difficulty, we propose to employ robust estimators of the center of the probability distribution of $\boldsymbol{\psi}$.

This robust estimation depends on the nature of the parametrization of the ellipse to be considered. As noted in (Rosin, 1993b), the algebraic parametrization of the ellipse is not adequate for accumulation of multiple fits. In this work, an alternative parametrization of the ellipse is proposed for this purpose:

$$\boldsymbol{\psi} = (\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma) = (\psi_1, \psi_2, \psi_3, \psi_4, \psi_5) \tag{10.2}$$

where (α_1, α_2) are the coordinates of the first focus of the ellipse, (β_1, β_2) are the coordinates of the second focus, and γ is the sum of the distances from each point of the ellipse to (α_1, α_2) and (β_1, β_2) . This way, the ellipse is defined as the set of points $\mathbf{x} \in \mathbb{R}^2$ which satisfy the following equation:

$$\|(\alpha_1, \alpha_2) - \mathbf{x}\| + \|(\beta_1, \beta_2) - \mathbf{x}\| = \gamma \tag{10.3}$$

where $\|\cdot\|$ stands for the Euclidean norm. It must be highlighted that (10.2) represents an ellipse if and only if the following condition holds:

$$\|(\alpha_1, \alpha_2) - (\beta_1, \beta_2)\| < \gamma \tag{10.4}$$

i.e. the focal distance is smaller than the sum of distances parameter.

Let us consider K random realizations of the probability distribution of $\boldsymbol{\psi}$, which are associated to K subsets of size N uniformly drawn without replacement from $\mathcal{S}$, $Q = \{\boldsymbol{\psi}_j : j \in \{1, ..., K\}\}$. The fact that (α_1, α_2) and (β_1, β_2) are points on the plane suggests a robust procedure to estimate the center of the distribution of $\boldsymbol{\psi}$. The robust estimates of the coordinates of the two foci are computed as follows:

$$(\hat{\alpha}_1, \hat{\alpha}_2) = \arg \min_{(\alpha_1, \alpha_2) \in \mathbb{R}^2} \sum_{j=1}^{K} \|(\alpha_1, \alpha_2) - (\psi_{j,1}, \psi_{j,2})\| \tag{10.5}$$

$$\left(\hat{\beta}_1, \hat{\beta}_2\right) = \arg \min_{(\beta_1, \beta_2) \in \mathbb{R}^2} \sum_{j=1}^{K} \|(\beta_1, \beta_2) - (\psi_{j,3}, \psi_{j,4})\| \tag{10.6}$$

that is, $(\hat{\alpha}_1, \hat{\alpha}_2)$ is the spatial ℓ_1 median of the first estimated foci $(\psi_{j,1}, \psi_{j,2})$, while $\left(\hat{\beta}_1, \hat{\beta}_2\right)$ is the ℓ_1 median of the second estimated foci $(\psi_{j,3}, \psi_{j,4})$. The robustness of the estimation comes from the fact that the sum of Euclidean distances is considered in (10.5) and (10.6), rather than the sum of the squares of the Euclidean distances. The latter choice would have led us to the mean of the sets of estimated foci, and the mean is known to be non robust (Maronna and Martin, 2019).

The estimate $\hat{\gamma}$ of the sum of distances parameter can also be robustly computed by the univariate median:

$$\hat{\gamma} = \arg\min_{\gamma \in \mathbb{R}} \sum_{j=1}^{K} |\gamma - \psi_{j,5}| \tag{10.7}$$

where $|\cdot|$ stands for the absolute value of a real number.

As done in (10.5) and (10.6), the selection of a robust estimator ensures that a small quantity of bad fits $\boldsymbol{\psi}_j$ does not affect the estimation too severely. A tentative estimation can be built by joining the results of (10.5), (10.6) and (10.7):

$$\hat{\boldsymbol{\psi}} = \left(\hat{\alpha}_1, \hat{\alpha}_2, \hat{\beta}_1, \hat{\beta}_2, \hat{\gamma}\right) \tag{10.8}$$

An additional postprocessing stage is proposed in order to filter out the deleterious effect of the bad fits. First of all, the Euclidean distance in the natural parameter space is computed between the tentative estimation and the fits:

$$d_j = \left\| \hat{\boldsymbol{\psi}} - \boldsymbol{\psi}_j \right\| \tag{10.9}$$

where $j \in \{1, ..., K\}$. Then the q-th quantile ω of the distances d_j is computed:

$$P\left(d < \omega\right) = q \tag{10.10}$$

where $q \in (0, 1)$ is a tunable parameter. After that, the final estimation is obtained as a robust estimate based on the best fits according to the distance d:

$$(\tilde{\alpha}_1, \tilde{\alpha}_2) = \arg \min_{(\alpha_1, \alpha_2) \in \mathbb{R}^2} \sum_{d_j < \rho} \|(\alpha_1, \alpha_2) - (\psi_{j,1}, \psi_{j,2})\| \tag{10.11}$$

$$\left(\tilde{\beta}_1, \tilde{\beta}_2\right) = \arg \min_{(\beta_1, \beta_2) \in \mathbb{R}^2} \sum_{d_j < \rho} \|(\beta_1, \beta_2) - (\psi_{j,3}, \psi_{j,4})\| \tag{10.12}$$

$$\tilde{\gamma} = \arg\min_{\gamma \in \mathbb{R}} \sum_{d_j < \rho} |\gamma - \psi_{j,5}| \tag{10.13}$$

$$\tilde{\boldsymbol{\psi}} = \left(\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\beta}_1, \tilde{\beta}_2, \tilde{\gamma}\right) \tag{10.14}$$

The proposed algorithm reads as follows:

1. For values of j between 1 and K, repeat steps 2 and 3.
2. Draw uniformly at random a subset T of the training set $\mathcal{S}$, with replacement.
3. Run the base ellipse fitting algorithm φ with input T to obtain the output ellipse fit $\boldsymbol{\psi}_j$.
4. Apply equations (10.5), (10.6) and (10.7) to obtain the tentative estimation $\hat{\boldsymbol{\psi}} = \left(\hat{\alpha}_1, \hat{\alpha}_2, \hat{\beta}_1, \hat{\beta}_2, \hat{\gamma}\right)$.
5. Compute the distances in the parameter space by (10.9) for all the fits.
6. Apply equations (10.11), (10.12) and (10.13) to obtain the final estimation $\tilde{\boldsymbol{\psi}} = \left(\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\beta}_1, \tilde{\beta}_2, \tilde{\gamma}\right)$.
7. If $\tilde{\boldsymbol{\psi}}$ corresponds to an ellipse as described by (10.4), then return $\tilde{\boldsymbol{\psi}}$ and halt. Otherwise, run a backup ellipse fitting algorithm with input $\mathcal{S}$, then return the obtained output and halt.

The backup procedure of step 7 is included because sometimes the tentative estimation does not describe an ellipse. This might happen when the training set $\mathcal{S}$ does not correspond to an ellipse, or when the individual fits $\boldsymbol{\psi}_j$ have a poor quality. It is worth noting that circles, as a special case of ellipses, can be detected equally well as non circular ellipses. For a circle we have that the two foci (α_1, α_2) and (β_1, β_2) are the same. This means that the estimated foci of the base ellipse fits $(\psi_{j,1}, \psi_{j,2})$ and $(\psi_{j,3}, \psi_{j,4})$ concentrate around the center of the circle. Consequently the final estimations of both foci $(\tilde{\alpha}_1, \tilde{\alpha}_2)$ and $\left(\tilde{\beta}_1, \tilde{\beta}_2\right)$ are close to the center of the circle, which is the correct solution. Figure 10.1 depicts an example of circle fitting. It can be observed that both estimated foci coincide up to a distance around 10^{-9}.

10.3.2 Robustness analysis

In order to assess the robustness of the SAREfit algorithm against outliers, a formal analysis of the expected number of 'bad' ellipses generated in step 3 of the algorithm is carried out in this subsection. This study is done with respect to the fraction of outliers in the training dataset and the robustness of the baseline algorithm. In other words, in this subsection, it is analyzed how many outlying input points can exist in the training set without preventing our approach to yield a reasonable output ellipse fit. Let us remember that the breakdown point of an estimator is the fraction of outlying input samples that must exist in order to make the estimator produce a broken estimation, i.e. an estimation with too large output parameters. On the other hand,

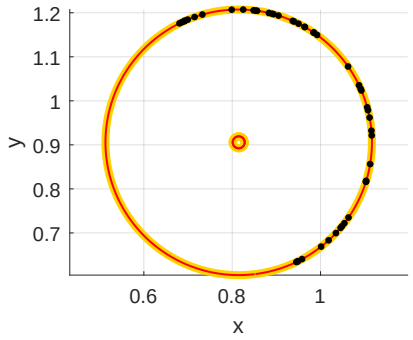


Figure 10.1: Example of the SAREfit execution with a perfect circle (degenerate case).

the q-th quantile of a distribution is the value that leaves a fraction q of the samples below the quantile.

Let us assume that the baseline algorithm has a breakdown point of $\xi \in [0, 1]$. That is, the baseline algorithm fails in the fit of the ellipse whenever the proportion of outliers in the input data to the baseline algorithm is higher than ξ. Also, let $\zeta \in [0, 1]$ be the proportion of outlying (bad) samples contained in the training set S. Thus, the number of outlying (bad) samples B in a subset T with cardinal N of the training set S follows a binomial distribution:

$$B \sim Bi(n = N, p = \zeta) \tag{10.15}$$

where N is the number of samples in a subset T to be provided as input to the baseline algorithm.

The baseline algorithm will produce a bad output ellipse whenever the number of outlying samples in its input subset T is between ξN and N. Next, the probability of having between ξN and N outlying (bad) samples in a subset T of size N of the training set S can be estimated computing the sum of the probabilities of achieving between ξN and N successes with the above binomial probability distribution:

$$P(\xi N \leq B \leq N) = P(B \geq \xi N) \tag{10.16}$$

For $\xi N \leq N$, upper bounds for the lower tail of the distribution function can be derived. Equivalently, the upper tail bound is derived directly from the Chernoff bound (Arratia and Gordon, 1989):

$$P(B \geq \xi N) \leq \exp\left(-N \cdot D(\xi \| \zeta)\right) \quad \text{if } \zeta < \xi < 1 \tag{10.17}$$

where $D(t\|s)$ denotes the relative entropy between the $Bernoulli(t)$ and

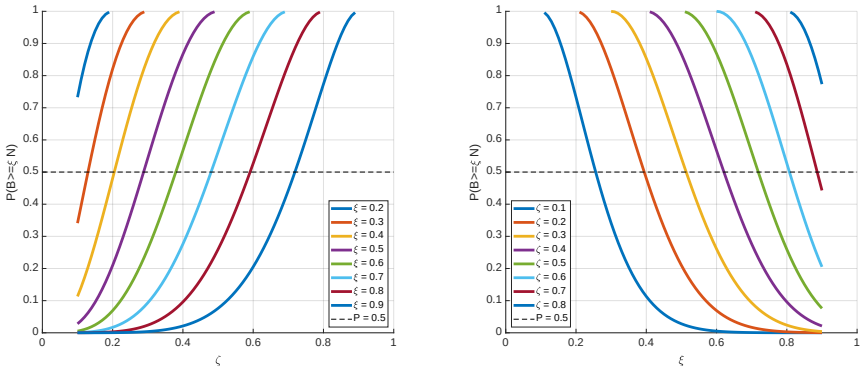


Figure 10.2: Graphical analysis of the robustness of SAREfit method. Values of ζ and ξ are sampled in the interval $(0, 1)$ and Eq. (10.19) is evaluated. Horizontal dashed black line represents the breakdown point of the median.

$Bernoulli(s)$ distribution and is obtained as:

$$D(t\|s) = t \log \frac{t}{s} + (1-t) \log \frac{1-t}{1-s} \tag{10.18}$$

Combining Equations (10.17) and (10.18), the following bound is derived:

$$P(B \geq \xi N) \leq \left(\frac{\zeta}{\xi}\right)^{\xi N} \left(\frac{1-\zeta}{1-\xi}\right)^{(1-\xi)N} \quad \text{if } \zeta < \xi < 1 \tag{10.19}$$

Equation (10.19) gives an upper bound of the probability $P(B \geq \xi N)$ that a bad ellipse is produced as the output of an execution of the baseline algorithm. Step 4 of our proposed algorithm (Subsection 10.3.1) employs medians to obtain the tentative estimation of the ellipse by computing medians of the parameters of the output ellipses coming from the baseline algorithm. Since the median has a breakdown point of 0.5, in order to avoid that step 4 of the algorithm generates a bad ellipse, it should be ensured that $P(B \geq \xi N) < 0.5$. From Equation (10.19), this can be guaranteed if the following condition holds:

$$\left(\frac{\zeta}{\xi}\right)^{\xi N} \left(\frac{1-\zeta}{1-\xi}\right)^{(1-\xi)N} < 0.5 \quad \text{if } \zeta < \xi < 1 \tag{10.20}$$

Therefore, values of ζ and ξ which fulfill Equation (10.20) should be employed in order to ensure the quality of the final ellipse fit.

In Figure 10.2 the representation of Equation (10.19) is depicted for a variety of values of ζ and ξ between 0 and 1. When ζ is increased, i.e. there are more outliers in the dataset, the baseline algorithm tolerates a higher proportion of bad samples before reaching the breakdown point of

0.5. Likewise, the second graphic shows that if the tolerance of the baseline algorithm to cope with many outlying points is good enough, then the whole dataset S may contain a larger number of outliers while our proposal still works fine since there are more values of ζ below $P = 0.5$.

10.4 Experimental Results

A set of experiments were designed in order to assess the robustness of our method. First, the evaluation metrics are described in Subsection 10.4.1 and the parameter optimization procedure is summarized in Subsection 10.4.2. The results on synthetic and real data samples are discussed in Subsections 10.4.3-10.4.6. Finally, the problem of multiple ellipse fitting is discussed in Subsection 10.4.7.

10.4.1 Evaluation method

The proposed method, named as SAREfit[1], was compared with other state-of art algorithms for ellipse fitting, i.e. Muñoz (Muñoz-Pérez et al., 2014), Taubin (Taubin, 1991), Szpak (Szpak et al., 2015), Fitzgibbon (Fitzgibbon et al., 1996), Halir&Flusser (Halir and Flusser, 1998), Rosin (Rosin, 1999), Prasad (Prasad et al., 2013), Köning (Köning et al., 2014), Liang (Liang et al., 2013) and Sobhani (Sobhani et al., 2017) methods. Default parameter values were used for all the methods, with the exception of Liang and Sobhani where their sparsity parameter λ was tuned for our experiments because their proposed value does not fit with our experimental setup. $\lambda = 0.2$ was chosen for both methods.

We have considered the use of performance profiles (Dolan and Moré, 2002) with the intention of condensing the results of numerous tests. The aim of this representation is to build a probability cumulative distribution that measures the performance of each method. Thus, the best method will be the one with the nearest probability curve to the left-upper corner of the figure axes. A detailed description of this representation can be found in the literature.

Four different error measures were considered in order to assess the robustness of our method compared with the competitors:

1. Natural error: the natural parametrization (10.2) of the fitted ellipse is computed and then it is compared with the true values as

$$NaturalError = \sqrt{\sum_{i=1}^{5}(\boldsymbol{\psi}^i - \tilde{\boldsymbol{\psi}}^i)^2} \tag{10.21}$$

[1]The source code and demo of the proposed approach will be published in case of acceptance.

2. Algebraic error: the algebraic parametrization of the ellipse is considered, $\boldsymbol{\xi} = (A, B, C, D, E, F)$ associated to the general equation of a conic section. The true and predicted parameters are normalized and the error is computed as

$$AlgebraicError = \sqrt{\sum_{i=1}^{5} \left(\frac{\boldsymbol{\xi}^i}{||\boldsymbol{\xi}||} - \frac{\tilde{\boldsymbol{\xi}}^i}{||\tilde{\boldsymbol{\xi}}||} \right)^2} \tag{10.22}$$

3. Geometric errors: given the geometric parametrization $\boldsymbol{\zeta} = (x, y, a, b, \theta)$, where $\mathbf{c} = (x, y) \in \mathbb{R}^2$ is the center of the ellipse, a is the half length of the major axis, b is the half length of the minor axis, $a \geq b > 0$, and $\theta \in [0, \pi]$ is the angle of tilt, we computed the errors of the center, angle, major and minor semiaxes and area of the ellipse:

$$\epsilon_c = \|\mathbf{c} - \tilde{\mathbf{c}}\|,\ \epsilon_a = |a - \tilde{a}|,\ \epsilon_b = \left|b - \tilde{b}\right|,\ \epsilon_A = \left|A - \tilde{A}\right|,\ \epsilon_\theta = \left|\theta - \tilde{\theta}\right| \tag{10.23}$$

4. Root Mean Squared Orthogonal (RMSO) error: a set of M points belonging to the true ellipse are projected to the estimated ellipse in order to compute their orthogonal distances o_i (Zhang, 1997). Then, the root mean squared error of those measures is calculated as:

$$RMSOError = \sqrt{\frac{1}{M} \sum_{i=1}^{M} o_i^2} \tag{10.24}$$

The use of more than one performance measure of different nature is a good way to assess the quality of a new methodology. Depending on the application for which the fitting method will be used, the principal error measure should be different. Our algorithm is based on the natural parametrization, so the natural error is a bit more biased because most of the competitors use the algebraic or geometric parametrization. Nevertheless, when the aim of the fit is to obtain a good precision on the focal points, this error has to be the reference. On the other hand, algebraic and geometric errors are correlated, and it is for interest when we want to achieve a correct orientation and center of the ellipse, for example. Finally, the RMSO error is a general measure that can be used to compare different methods irrespective of their fitting criterion and the quality of the fitted ellipses (Kurt and Arslan, 2019). In order to overcome problems with algebraic errors, we can make use of the RMSO error which is not associated with any parametrization, is invariant to transformations in Euclidean space and has less high curvature bias. It is also suitable when the true ellipse is not provided although might be affected by the presence of outliers.

Table 10.1: Parameters tested for the SAREfit algorithm and the optimal results according to the Natural, Algebraic and RMSO errors. The selected parameters are in bold.

Parameter	Tested values	Nat. Er.	Alg.Er.	RMSO Er.
s_T	{0.1,...,1} (step 0.05)	**0.15**	0.15	0.70
K	{15,...,150} (step 15)	**90**	90	15
q	{0,0.05,0.1,...,1} (step 10)	**0.10**	0.10	0

All experiments were carried out on a 64-bit PC, with an Intel Core i7-4790, 3.6GHz CPU, 32 GB RAM and standard hardware. MATLAB R2018b was used to implement the proposed method as well as to perform the comparisons with the competing methods, with no use of GPU resources.

10.4.2 Parameter fitting

SAREfit model is based on an ellipse fitting algorithm φ and a backup algorithm too, as described in the previous section. Due to its extreme simplicity, Halir&Flusser's method was chosen to be this algorithm φ, and Fitzgibbon to be the backup method. Both of them with their default parameters. Nevertheless, SAREfit includes its own parameters that need to be fitted:

- Subsampling factor (s_T): is the proportion of the input samples considered in the fitting procedure. size of the set S creates the input set T, as it is described in Section 10.3.
- Ensemble size (K): the number of distributions created from the whole samples, i.e., the number of ellipses fitted by the algorithm using different training sets.
- Quantile (q): the q-th quantile value used to filter the fits according to the natural error.

In Table 10.1 both the set of tested parameters and those found to be the best ones are summarized according to the natural, algebraic and RMSO errors. All the possible combinations of parameters were run in order to find a global minimum. 100 different initializations were executed incorporating between 5% and 20% of outliers randomly, and the average errors were calculated to have a precise estimation. $q = 0$ means that the best ellipse based on the error with respect to the tentative estimation $\hat{\boldsymbol{\psi}}$ is chosen. The best configuration was exactly the same for two types of errors. $s_T = 0.15$ indicates that 15% of the input points are enough to achieve good results. This can be interpreted as a way to remove outlier points from the training dataset. $K = 90$ fits seems to be enough to achieve stability across different initializations. Finally, the post-selection of the best 10% of ellipses with respect to

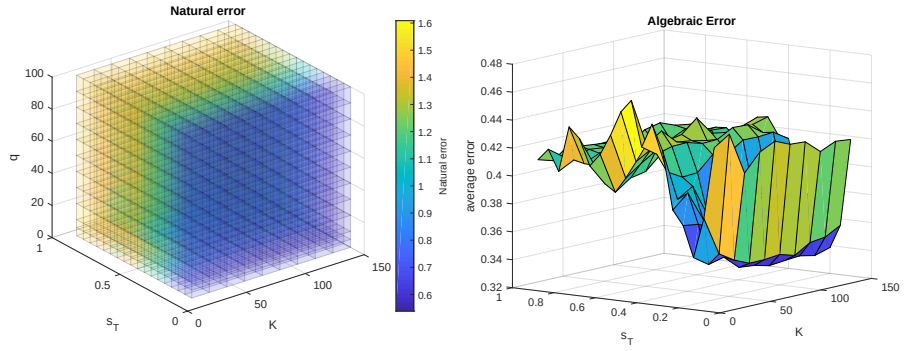


Figure 10.3: Parameter optimization based on the average along 100 executions. Left: 4D representation of the Natural error obtained in the simulations (the darkest the better). Right: Algebraic errors varying the subsampling factor and the ensemble size, and fixing $q = 0.10$.

the first estimation assures improvement of the outcome. These estimated parameters were used for both synthetic and real experiments. Figure 10.3 depicts the detailed results of the optimization. The cube represents all possible combinations of the parameters, whose outcomes are plotted following the color bar. The best results are achieved for small values of Subsampling factors, which confirms our supposition about the good performance with the presence of outliers. The right side image represents one slice of the cube ($q = 0.10$), so the tendency of the other two parameters can be determined. We can observe a clear minimum at $s_T = 0.15$ and a stabilization of the error when the number of fits is increased.

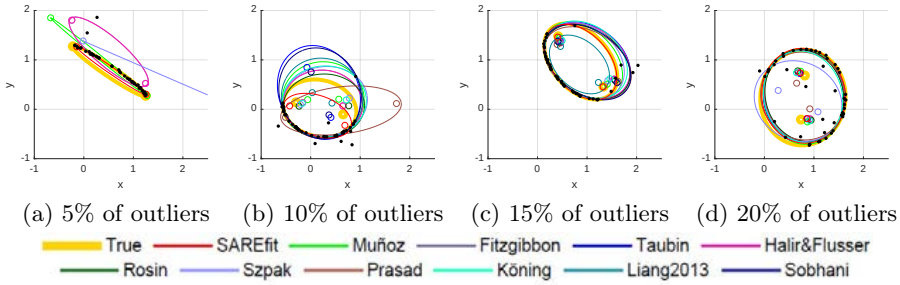
10.4.3 Synthetic data

The first experiments deal with artificial datasets generated as follows:

1. Firstly, the center, the major and minor axes of a ellipse are chosen at random uniformly: $x, y \sim U(0, 1)$, $a \sim U(0.2, 1)$, $b \sim U(0.1, 1)$. The major and minor axes have different ranges to avoid degenerated ellipses. The tilt angle is also chosen uniformly $\theta \sim U\left(\frac{-\pi}{2}, \frac{\pi}{2}\right)$.

2. Secondly, $N = 50$ sample points are generated on the canonical coordinate system: $(a\cos\phi, b\sin\phi) \in \mathbb{R}^2$, where $\phi \sim U(\phi_s, \phi_e)$, $\phi_s, \phi_e \sim U(-\pi, \pi)$, and satisfying that $\phi_e - \phi_s > 1$ in order to avoid the prediction of degenerated ellipses caused by datasets with small curvature. The random choice of the starting and ending angles combined with this constraint allows having different levels of occlusion in the training points of the ellipse.

3. Finally, 1% of normally distributed Gaussian noise is added and a specific percentage of the points are modified to convert them to outliers.



(a) 5% of outliers (b) 10% of outliers (c) 15% of outliers (d) 20% of outliers

Figure 10.4: Graphical comparison using synthetic data generation. Four different initializations and their solutions are shown. The black points are the training samples and the true ellipse is plotted as a thick solid curve, while the fitted ellipses are shown as a narrow solid curve.

4. In addition to this, 1000 test points on the true ellipse are generated to carry out the computation of the RMSO error.

Figure 10.4 presents four examples of synthetic ellipses whose training samples contain 5%, 10%, 15% and 20% of anomalous points respectively. Figure 10.4a shows almost a degenerated ellipse, and most of the competing methods failed in the fit. Muñoz, Halir&Flusser and Szpak methods achieved to output an ellipse, but SAREfit was the only one able to produce an ellipse with very similar characteristics to the ground truth, with only a small displacement. Second and third examples (Figures 10.4b-10.4c) also have a high level of occlusion (around 50%), and the number of outliers was increased. In both cases, all methods had a better outcome, although most of them were affected by the presence of outliers. Again, the most accurate prediction was the one generated by SAREfit, followed by Rosin and Muñoz methods. At last, an example with 20% of outliers and less occlusion is shown in Figure 10.4d. All the algorithms approximately fitted the ellipse with the exception of Liang algorithm, and our proposal is the only method which completely overwrites the edge of the ellipse.

Next, batches of 1000 different random initializations and with random occlusion were run varying the percentage of outliers between batches, and the results are summarized in Figure 10.5 using the performance profiles described before. Note that the best method is the one that first reaches probability 1. Figures 10.5a-10.5f contains the outcomes of 5% and 10% of outliers. There are six methods that never attain to solve all the possible configurations posed in the batch: Taubin, Rosin, Prasad, Köning, Liang, and Sobhani. This means that they are very unstable if the input samples do not have a refined shape of an ellipse. There is more variability with the rest of the methods depending on the metric. In terms of Natural and Algebraic error, SAREfit is clearly the best method, solving always all the

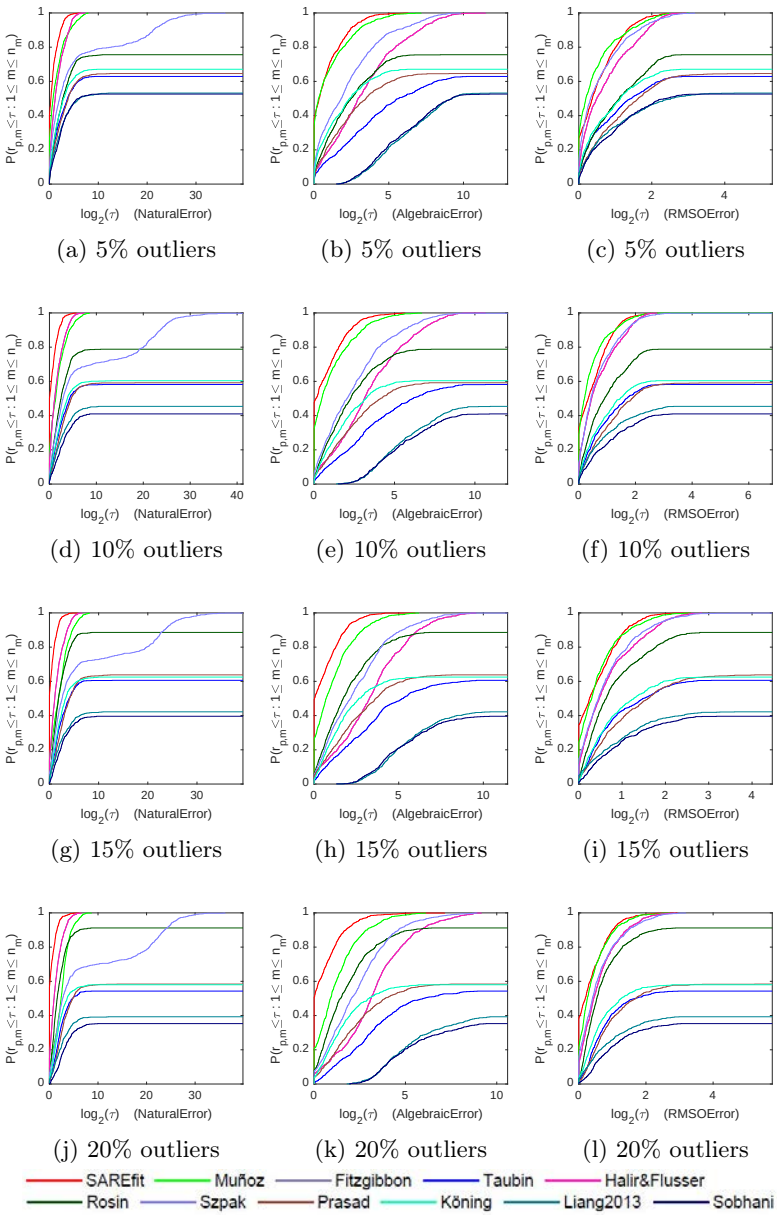


Figure 10.5: Performance profiles of synthetic experiments (the closer to the upper left corner, the better). Natural, Algebraic and RMSO errors are analyzed with 5%, 10%, 15% and 20% of outliers. X axis shows the factor of the best possible ratio in a logarithmic scale and Y axis represents the probability cumulative distribution.

fits with less error than the competitors. Moreover, our proposal is also good using the RMSO measure as the hard training datasets ($\tau > 2$) are solved with less error, i.e. when the quantity of outliers is incremented, the results were more robust in favor of SAREfit. Muñoz and Szpak algorithms also perform well for the Natural and RMSO errors. Figures 10.5g-10.5l show the results of 15% and 20% of anomalous points. Here, Rosin method improved its results reaching almost 90% of the ellipse fits. On the other side, the positive differences observed with respect to Muñoz and Szpak methods is incremented and the gap based on RMSO error is clearly reduced. The case of Liang and Sobhani methods is very particular because one expects a better performance. However, these methods model the sparsity of the samples and they work well under controlled situations, such as the absence of occlusion and large numbers of samples. Unlike SAREfit, this makes both methods inappropriate for extreme configurations. Nevertheless, when the number of outliers is fixed and there is no occlusion (see Subsection 10.4.5) they improve although they are also affected by eccentricity.

Figure 10.6 sums up the geometric errors for 10% of outliers. The mean and median values in the box plots are shown as a gray circle and a line, respectively. A small pattern can be found in each feature with respect to our proposal: SAREfit has located its average and median below the ones obtained by the competing methods, although there is a bit more dispersion across the fits. This means that most of the input samples are fitted better, although sometimes the differences are larger. The worse method is Szpak, showing a great dispersion caused by the generation of incorrect ellipses. SAREfit and Muñoz methods are the best ones focusing on the center, area and angle, followed by Fitzgibbon. Köning method stands out for its nice performance in center, semiaxes, and area but it has problems to fit adequately the angle of the ellipse, which might be caused by the outliers. Liang and Sobhani methods yield adequate values for the center and area, but the size and orientation of the fitted ellipse are imprecise, affecting all error measures.

The outcomes for 20% of outliers are presented in Figure 10.7, where the differences between methods are more obvious. Fitzgibbon, Halir&Fluser, and SAREfit returned the lowest errors, especially for the center, major semiaxis, and area. Muñoz still worked fine in terms of angle and minor semiaxis but its median and mean value are larger for the other features, which are in consonance with the gap found in the performance profiles. Prasad method has the third best results focusing on angle orientation and minor semiaxis, but this is not enough to obtain a good fit, as it is observed with the other metrics. Liang and Sobhani methods are still behind most of the methods although their error range is small, which shows some level of stability. The main problem of most methods is the orientation of the fitted ellipse, which causes major differences in terms of performance.

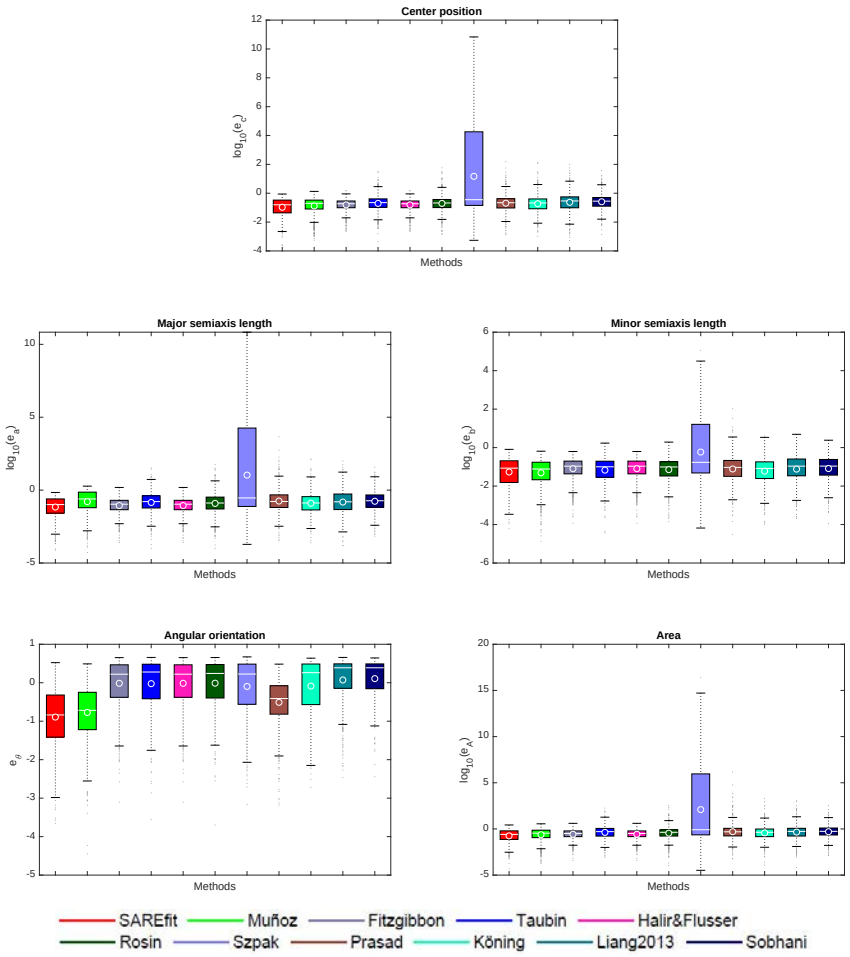


Figure 10.6: Major and minor axes, center, angle and area box plots comparisons. Geometric errors are analyzed with 10% of outliers. X axis shows the factor of the best possible ratio in a logarithmic scale and Y axis represents the probability cumulative distribution.

In addition to the previous tests, an analysis of the upper limit of outliers was carried out to check how the competing methods deal with anomalous samples. Therefore, a collection of 1000 runs without occlusion was executed for different percentages of outliers, ranging from 0% to 50%. Note that higher values of outliers might be considered as extreme noise level that modifies the original elliptical shape of the data points. In the first three plots of Figure 10.8, the median errors obtained for each measure are shown. It is worth noting that the higher level of outliers the more wrong fits are obtained by most of the methods, as shown in the last plot. These bad outputs were discarded in the computation of the median, making smoother the error curves from 30% onwards. Only SAREfit, Muñoz, Fitzgibbon and Halir&Flusser methods do not fail in the fit, which makes them very stable algorithms. Algebraic measure reveals that Taubin, Fitzgibbon, Sobhani and Liang methods are completely unstable. The break point of most of the algorithms is placed at 10% of outliers.

10.4.4 Computational efficiency

The performance of each method in terms of computation time was first analyzed. The mean, median, standard deviation and interquartile range values of 1000 runs are presented in Table 10.2. In general terms, fitting an ellipse does not take a long time. Most of the methods compute a fit in less than one millisecond, with the exception of Szpak, Muñoz, Köning, Liang, and SAREfit. Liang and Szpak methods are the slowest ones, while their irregular performance does not justify this computational need. Taking into account that SAREfit employs Halir&Flusser method as a baseline carrying out 90 executions, 0.0116 seconds for the whole procedure is an acceptable execution time for any ellipse fitting method, being adequate for its use in real-time applications. Moreover, it has to be noted that the presence of a cache may improve the performance of the algorithm, by making the execution of the loops faster.

From a theoretical point of view, SAREfit requires $O(K)$ iterations of its main loop, where the computational complexity of each iteration is the one of the baseline algorithm, which is linear with respect to the number of input samples (Halir and Flusser, 1998), $O(N)$. Therefore, the complexity of our proposed algorithm is $O(KN + N \log N)$, where the last term comes from the computation of the spatial median. Compared with the competitors (see Table 10.2), our proposal has an acceptable complexity. The most recent works are highly dependent from the number of points, $O(N^2)$ and above, being $O(N \log N)$ more efficient than them with higher quality performance.

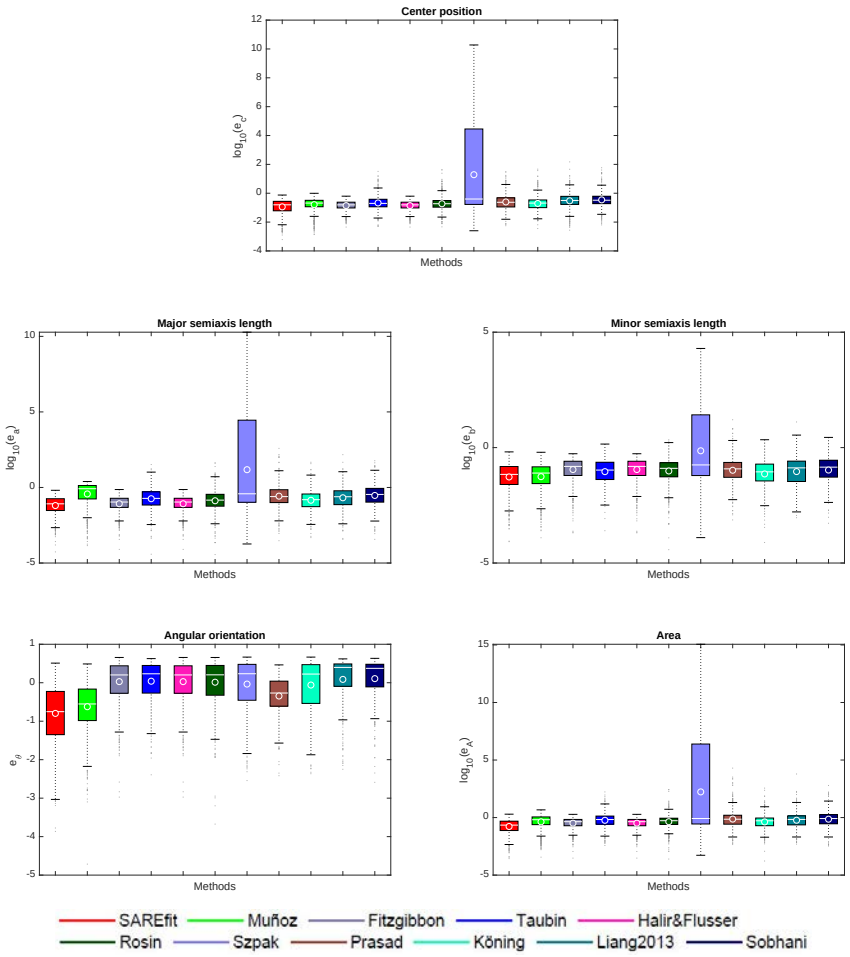


Figure 10.7: Major and minor axes, center, angle and area box plots comparisons. Geometric errors are analyzed with 20% of outliers. X axis shows the factor of the best possible ratio in a logarithmic scale and Y axis represents the probability cumulative distribution.

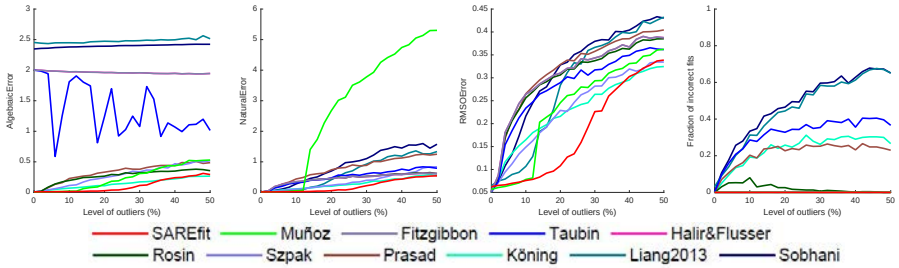


Figure 10.8: Evolution of the performance of the methods varying the level of outliers (0-50%). Median values of the algebraic, natural and RMSO errors over 1000 runs are represented. Las t figure counts the number of not successful fits.

Table 10.2: CPU times and complexity comparison of the methods using synthetic data. Mean, median, standard deviation and interquartile range computed among 1000 runs.

Method	**CPU time (s)**				**Complexity**
	Mean	**Median**	**Std**	**IQR**	
SAREfit	0.0116	0.0108	0.0019	0.0018	$O(KN + N \log N)$
Muñoz	0.0066	0.0060	0.0017	0.0006	$O(N \log N)$
Fitzgibbon	0.0002	0.0002	0.0001	0.0001	$O(N)$
Taubin	0.0002	0.0002	0.0002	0.0000	$O(6N + 72) \sim O(N)$
Halir&Flusser	0.0001	0.0001	0.0001	0.0000	$O(N)$
Rosin	0.0002	0.0001	0.0002	0.0000	$O(N)$
Szpak	0.0189	0.0123	0.0231	0.0107	$O(200N) \sim O(N)$
Prasad	0.0001	0.0001	0.0002	0.0000	$O(30N) \sim O(N)$
Köning	0.0073	0.0070	0.0023	0.0010	$O(N^3)$
Liang	0.3901	0.3839	0.0306	0.0223	$O((2N + 1)^{3.5} + N^2)$
Sobhani	0.0026	0.0024	0.0005	0.0003	$O(N^2)$

10.4.5 Occlusion analysis

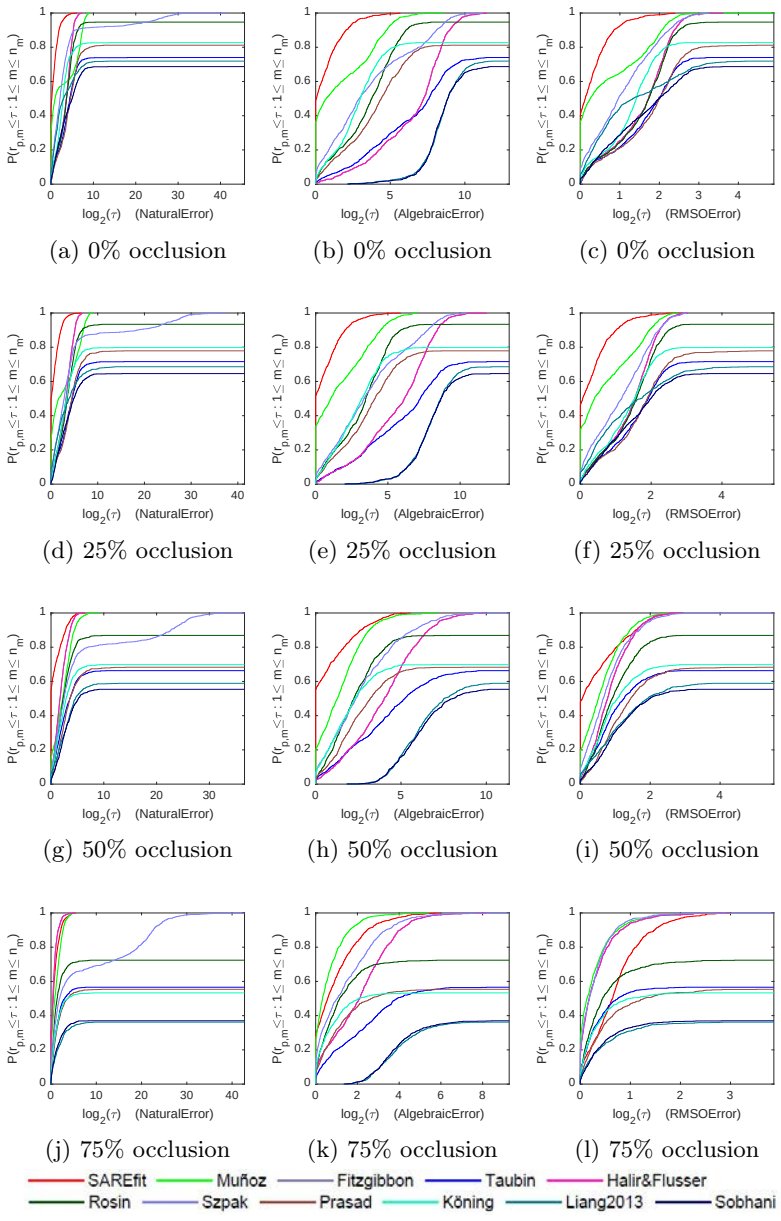
This Subsection intends to evaluate in detail the performance of each method with the presence of low, medium and high levels of occlusion. For this purpose, again a batch of 1000 runs with 0%, 25%, 50% and 75% of occlusion was executed and error measurements were computed. The procedure was as it follows: starting angle of the generated ellipse is selected from $\phi_s \sim U(-\pi, \pi)$. Then, the ending angle is computed as $\phi_e = \phi_s + occl \cdot 2\pi$, where *occl* represents the occlusion level in the range $[0, 1]$. The fraction of outliers was chosen at random from the uniform distribution between 0% and 20%.

Figure 10.9 presents the performance profiles of the Natural, Algebraic and RMSO errors The general tendency is the more occlusion is present in the training samples, the less is the quality of the fit for all methods. Nevertheless, in most cases, SAREfit method yields the best result. When the shape of the ellipse is complete or only a quarter is missing, our proposal is significantly better in terms of Algebraic and RMSO errors. Here, Liang and Sobhani methods perform much better. There are around 40% of ellipses that are fitted better than most of the methods, as the RMSO and Natural errors demonstrate. The comparisons with the Natural error are stretcher when the level of occlusion was increased. For 50% of occlusion, Szpak method become competitive for the natural error, and with Muñoz algorithm, the 1000 runs (probability 1) are solved with almost the same error as the obtained by SAREfit. When the tests are carried out with 75% of occlusion, Muñoz method is the best for the Algebraic error followed by our proposal, but being Szpak the second best in terms of orthogonal error and the first method for the Natural error.

An example of the outcomes with the presence of occlusion is depicted in Figure 10.10. For 25% the fit of our proposal is almost perfect although Muñoz method fails. Also, Szpak method is near to the true ellipse, shown in yellow. Köning, Fitzgibbon, Prasad, Liang, and Sobhani do not fit an ellipse. With 50% of occlusion, the competitors are clearly affected by the presence of outliers. Finally, with 75% of occlusion, most of the methods go through the ellipse points but fail in length of the major and minor semiaxes. SAREfit yields the most similar ellipse to the true one.

10.4.6 Real data

This subsection presents real examples of four images: two of them from the Caltech 256 dataset (Griffin et al., 2007), named as *Satellite dish* (image '169_0015') with size 448×336, and *Can* (image '195_0039') with size 247×350, and other two images of wheels captured by a normal camera, named as *Wheel 1* and *Wheel 2*, with sizes 2061×2891 and 2416×2024, respectively. Canny edge detector algorithm was used to randomly extract a set of 50 points from each image, using threshold parameter equal to 0.1



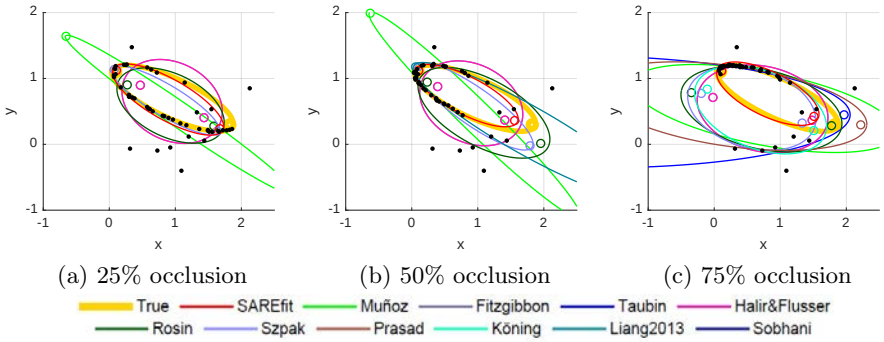
(a) 0% occlusion (b) 0% occlusion (c) 0% occlusion

(d) 25% occlusion (e) 25% occlusion (f) 25% occlusion

(g) 50% occlusion (h) 50% occlusion (i) 50% occlusion

(j) 75% occlusion (k) 75% occlusion (l) 75% occlusion

Figure 10.9: Performance profiles of the occlusion experiments (the closer to the upper left corner, the better). Natural, Algebraic and RMSO errors are analyzed with 0%, 25%, 50% and 75% of occlusion. X axis is shown in a logarithmic scale and Y axis represents the probability cumulative distribution.



(a) 25% occlusion (b) 50% occlusion (c) 75% occlusion

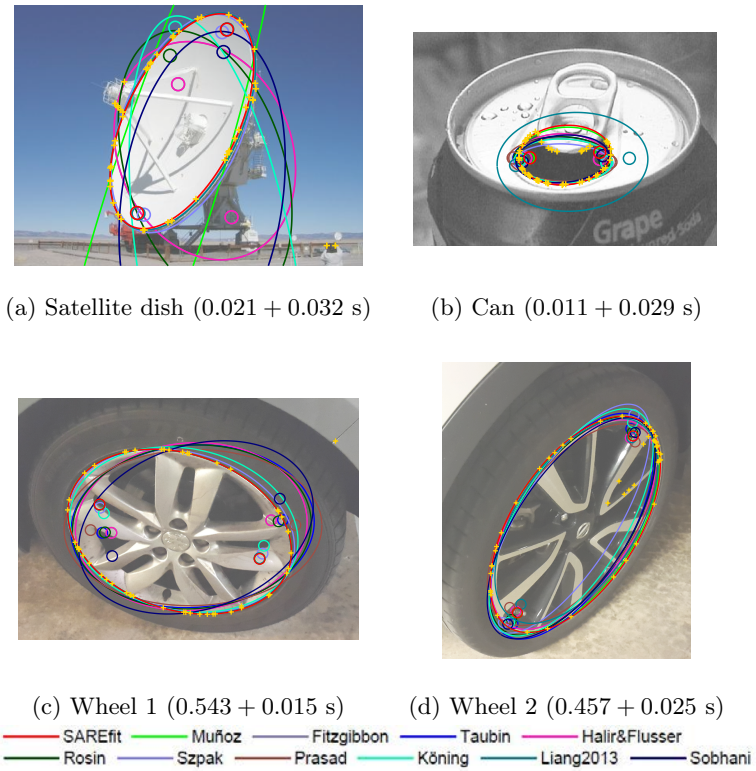
Figure 10.10: Example of the outcomes with different levels of occlusion. The black points are the training samples and the true ellipse is plotted as a thick solid curve, while the fitted ellipses by each method are shown as a narrow solid curve.

for the *Satellite dish*, and 0.9 for the other images. Resulting binary images were refined using the *imfill*, *imclearborder*, *im2bw* and *bwperim* MATLAB morphological functions with default parameters.

Figure 10.11 shows the fits generated by the eleven tested methods. The *Satellite dish* (Figure 10.11a) contains two extraneous points in the bottom right part of the image, as well as those located on the focus of the antenna, provoking that all methods failed in the fit except SAREfit and Szpak methods. The inferior part of the Szpak outcome falls outside the border of the satellite dish, being SAREfit more precise. In the case of the *Can* (Figure 10.11b), there is a group of points located on the superior part which hampers the fit processing. Muñoz and our method are the only ones that were able to find the real solution. Finally, Figures 10.11c-10.11d are examples with some outliers outside and inside the ellipse shape respectively. Here, the predictions are more accurate, although again SAREfit and Muñoz algorithms achieved the best fit. The fits of the other methods are varied: Szpak and Liang are precise in the first wheel but not good in the second, Köning method has a small deviation, and the rest of the methods are affected to a greater or lesser extent.

In order to have a quantitative point of view of these fits, we computed the RMSO error with respect to a set of 1000 points generated on the shape of the real ellipse. For that purpose, we first used the *Ellipse Labeling Tool*[2] to select five points and we solved the general equation of an ellipse to assure its precision overlaying the ellipse on the real image. After that, we were able to generate any point of this ellipse. Detailed results and a ranking

[2] https://sites.google.com/site/dilipprasad/Source-codes (accessed on June 10, 2019)



(a) Satellite dish ($0.021 + 0.032$ s) (b) Can ($0.011 + 0.029$ s)

(c) Wheel 1 ($0.543 + 0.015$ s) (d) Wheel 2 ($0.457 + 0.025$ s)

Figure 10.11: Graphical comparison of the performance of the tested methods using real data. The yellow crosses are the training samples, while the fitted ellipses by each method are shown as a narrow solid curve. Captions include processing time divided in two terms: the first one represents the image loading and point extraction, and the second is the SAREfit execution.

Table 10.3: Quantitative comparison of the methods using real data according to the RMSO error. 1000 test points on the ellipse shape of the object were generated and used to compute the errors. Those methods which did not fit the ellipse are represented as a dash. Rank points of each example are shown in brackets, as well as the total count.

Method	Satellite dish	Can	Wheel 1	Wheel 2	**Rank**
SAREfit	**1.1180 (1)**	**1.0361 (1)**	**2.1877 (1)**	**2.0712 (1)**	**4**
Muñoz	4.4465 (4)	1.1322 (2)	2.1148 (2)	2.6731 (2)	10
Fitzgibbon	5.2327 (7)	2.1789 (3)	9.0742 (7)	5.2885 (3)	20
Taubin	— (10.5)	2.4739 (7)	9.3524 (9)	5.9177 (7)	33.5
Halir&Flusser	5.2327 (8)	2.1789 (4)	9.0742 (6)	5.2885 (4)	22
Rosin	5.2089 (6)	2.4510 (6)	9.1103 (8)	5.7541 (5)	25
Szpak	1.8212 (3)	3.0478 (10)	3.3663 (4)	8.9530 (11)	28
Prasad	— (10.5)	2.7081 (9)	9.4619 (10)	6.5871 (8)	37.5
Köning	5.1459 (5)	2.6124 (8)	5.4467 (5)	6.9363 (10)	28
Liang	1.3139 (2)	5.0095 (11)	2.3093 (3)	5.9070 (6)	22
Sobhani	5.5837 (9)	2.3151 (5)	11.2278 (11)	6.6144 (9)	34

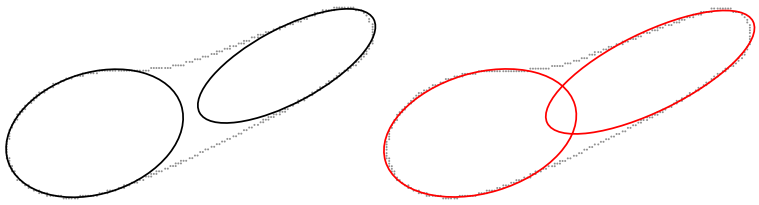
of methods are shown in Table 10.3. Results were ordered increasingly and their position determined the rank value. For all the images our method is the best and the error values are always very small, indicating that the fitted ellipse is very close to the real shape. The competing methods are rather unstable, as can be seen in the error values and the rank position. With 4 points, SAREfit is the winner, followed by Muñoz which is also robust achieving 10 points.

Finally, in the captions of the images, we have included the computation time needed to perform the ellipse fitting. These values are separated into two terms. The first represents the CPU time necessary to load the image and execute the Canny detection algorithm. This value is highly dependent on the size of the image. After that, the second term corresponds to the SAREfit execution time, whose values are quite similar to the CPU time analysis done before.

10.4.7 Multiple ellipse fitting

Multiple ellipse fitting is one of the main challenges of this field that has a variety of applications in biology, medicine, architecture, etc. Although SAREfit is developed only for single ellipse fitting. In this subsection, we address the case of multiple fits in the same cloud point.

For that purpose, a framework rather than design an algorithm for the problem could be used. Hui Li (Li, 2019) has recently proposed a four-step methodology to find all possible ellipses based on three simple techniques:



(a) Original method (b) SAREfit adapted method

Figure 10.12: Graphical comparison of the operation of the multiple ellipse fitting methods.

single ellipse fitting, anomaly detection, and clustering. Thus, its architecture allows to include any kind of single ellipse fitting algorithm. The source code was implemented in Mathematica and it was kindly provided by the author. Our proposal was connected with his code through MATLink library[3].

Figure 10.12 presents the results of both the original Li's method and the adapted one for a specific set of data samples. Both executions detect two ellipses and the final result depends on the quality of the fitting algorithm. In this example, SAREfit generates larger ellipses than the original method, which more closely follow the shape of the point cloud. The case of the Can example (Figure 10.11b) can be addressed with the same methodology. First, the point extraction step should preserve points of all the ellipse shapes present in the image and then this procedure will cluster, detect and fit the corresponding ellipses.

An advantage of our proposal is that can be easily integrated into any other kind of method for multiple fits where the ellipse fitting procedure is a separate component of the whole process.

10.5 Conclusions

In this section, we extract some conclusions about the results of this research. An ensemble method that is based on a natural parametrization has been proposed for ellipse fitting. It selects the best fits among a set of fits obtained by random subsampling of the available training set. The spatial median is used to combine the best fits into a single ellipse fit which is provided as the output of the procedure.

Stability across different initializations is found after reaching 90 ellipse fits, of which only the best 10% are selected. This fact also assures performance improvement. According to the outcomes of the tests, it is confirmed that the best results are achieved for small values of subsampling factors, 15%

[3] http://matlink.org/

of the input points are enough, reinforcing the idea of good performance under the presence of outliers. This is stated after performing experiments with different levels of outliers and occlusion and comparing the results for the Natural, Algebraic and RMSO errors, as well as individual Geometric errors. Although datasets with small curvature were dismissed to avoid degenerated ellipse as outcomes, the tests performed with an almost degenerated ellipse and 5% of outliers show that most competing methods failed in the fit, being SAREfit the only one to achieve an ellipse with similar characteristics to the true one. Unlike the competing methods, SAREfit solves all the fits with less error in terms of the Natural, Algebraic and RMSO errors. There is some variability depending on the metrics considered, but the combination of the spatial median with the post-processing refinement provides more precise natural and algebraic parameters. Regarding geometric parameters, most of the input sample sets are better fitted by SAREfit method than its competitors, especially focusing on the center, area and angle.

As a general conclusion, it can be said that SAREfit method outperforms the state-of-the-art of ellipse fitting methods, especially when outliers are present, measured in terms of three ellipse parametrizations and the closeness to the true ellipse through the orthogonal distances. Real data tests performed on several natural images also confirm such a statement. The accuracy of the estimation depends on the number of outliers that are present in the input set of points. This is determined by the ability of the procedure which provides the input set to our algorithm to overcome challenging effects such as illumination conditions and camera limitations. These data acquisition issues are outside the scope of our work, which deals with ellipse fitting only. A weakness of our procedure is that it relies on a base method so that gross estimation errors from the base method can affect the performance of our approach.

Our work is devoted to fit a single ellipse to a set of input points. In order to integrate our approach into a multiple ellipse detection system, a clustering subsystem should be added so as to provide our method with an input set of points that are expected to belong to a single ellipse. After our method produces an estimation of the ellipse parameters, the quality of the fit must be evaluated by a subsequent ellipse assessment subsystem, in order to ascertain whether the input set of points should be modified. This way, the segmentation of the overall set of points into ellipses would be refined over time. The advantage of integrating our approach into a multiple ellipse detection system is that its robustness allows obtaining a good fit of an ellipse even if the previous clustering subsystem wrongly introduces some points belonging to other ellipses into the input set of points for our algorithm.

Further works include the improvement of the ensemble technique. At this stage all base ellipse fits are equally likely to get into the ensemble and

they are given the same importance. Therefore, a possible enhancement of the proposal would be the selection of promising base fits for their inclusion in the ensemble, and the usage of weights to give more importance to those base fits which are more likely to be accurate. Additional enhancements might be attained by, using a combination of the error criteria in order to have a comprehensive evaluation of the base fits and the ensemble fits. Hopefully this would allow solving more difficult scenarios with high occlusion and higher levels of noise. The integration of SAREfit into an ellipse detection system would make it suitable to be applied to several areas of engineering, medicine or astronomy where multiple ellipses exist in the same dataset so that clustering of the sample points must be done at the same time that the individual ellipses are fitted.

Chapter 11

Robust fitting of ellipsoids with adaptive step size control

Controlling complexity is the essence of computer programming.

Brian Kernighan

Abstract: Fitting geometric or algebraic surfaces to 3D data is a pervasive problem in many fields of science and engineering. In particular, ellipsoids are some of the most employed features in computer graphics and sensor calibrations. They are also useful in pattern recognition, computer vision, body detection and electronic device design. Standard ellipsoid fitting techniques to solve this problem involve the minimization of squared errors. However, most of these procedures are sensitive to noise. Here, we propose a method based on the minimization of absolute errors. Although our algorithm is iterative, an adaptive step size is used to achieve a faster convergence. This leads to a substantial improvement in robustness against outlier data. The proposal is demonstrated with several computational examples which comprise synthetic data and real data from a 3D scanner and a stereo camera.

11.1 Introduction

The minimization of the sum of absolute errors (also called ℓ_1-norm minimization) has been used previously to fit surfaces in 3D. The method by Flöry and Hofer (Flöry and Hofer, 2010) is aimed to fit B-spline surfaces, so that the spline control points can be displaced slightly to deform the surface

locally in order to fit the local features of the point cloud. This requires the use of a foot point in the surface for each sample point, so that the distance of the sample to the surface is computed with respect to the foot point. Then one can use three (non squared) distance measures: (a) the distance between the foot point and the sample point, (b) the distance from the tangent plane at the foot point to the sample point, (c) a second-order approximant of the signed distance function, based on the local Frenet frame associated to the foot point. Depending on the chosen option the ℓ_1 distance minimization is carried out by second-order cone programming, linear programming, or the proximal bundle method. In the linear programming case, the introduction of auxiliary variables enables the reformulation of the absolute values into linear functions.

Our approach considers a much more restricted class of surfaces, namely ellipsoids. This means that the surface can not be deformed locally, i.e. there are no control points. Foot points are not necessary in our approach because the analytic form of the ellipsoid allows computing the deviation of the sample points from the surface without foot points. The objective function is non linear, so a specifically tuned version of gradient descent is used instead of linear programming. The role of the reformulation of the absolute value by auxiliary variables in Flöry and Hofer's method is accomplished in our method by the split of the point cloud into two sets, namely the interior points and the exterior points.

From the above it can be said that, although the kind of surfaces to be fitted and the methodology of our proposal differ significantly from Flöry and Hofer's, both take advantage of the robustness against outliers provided by ℓ_1 optimization.

This chapter is structured as follows. Section 11.2 presents previous works and the definition of earlier ellipsoid fitting methods. Subsequently, Section 11.3 describes the formulation of our method and Section 11.4 reports the obtained experimental results using synthetic and real data. Finally, Section 11.5 summarizes the main conclusions of this research.

11.2 Previous work

In literature, a wide range of fitting techniques have been applied depending on the type of distance used: algebraic or geometric.

11.2.1 Algebraic fitting methods

In (Li and Griffiths, 2004) it is shown that when $aJ - I^2 > 0$, Eq. (2.7) described in Chapter 2 must represent an ellipsoid if $a = 4$. Hence, some authors that use the least squares fitting problem based on algebraic distance must consider this constraint. In Direct Method (DM) (Li and Griffiths,

2004) the constraint confines the class of ellipsoids to fit to those whose smallest radius is at least half of the largest radius; in the Simple method, the ellipsoid is fitted by bisection search on a in the equation $aJ - I^2 > 0$.

Ying et al. (2012) work intend to find an algebraic method for multi-dimensional ellipsoid-specific fitting, rather than fitting general quadratic surfaces. They draw inspiration from a study on the ellipse-specific fitting approach proposed by Fitzgibbon et al. (Fitzgibbon et al., 1996) since it outperforms other approaches because it is robust, efficient, and easy to implement. However, they mention that using linear least-square techniques with a constraint whose degree is greater than 2 is very difficult to carry out.

Also, as it is well known, a quadratic surface in n-dimensional space is defined as the locus of the zeros of a quadratic polynomial. This quadratic polynomial may be written in a different notation as a real symmetric matrix of order $n + 1$ and an $(n + 1)$-vector. The leading $n \times n$ principal submatrix of the symmetric matrix is positive or opposite definite whenever the n-D quadratic surface is an n-D ellipsoid. Fortunately, to ensure that a matrix is positive or opposite definite, the most practical way may be to utilize semidefinite programming (SDP) as Calafiore proposed (Calafiore, 2002). However, in Calafiore's method, the run time is significantly long and memory is often exhausted when the number of fitted points is greater than several thousands. Hence, Ying et al. (Ying et al., 2012) proposed a fast algorithm by minimizing a new defined vector norm of the algebraic residual vector using SDP by decreasing the dimensions of the SDP problem.

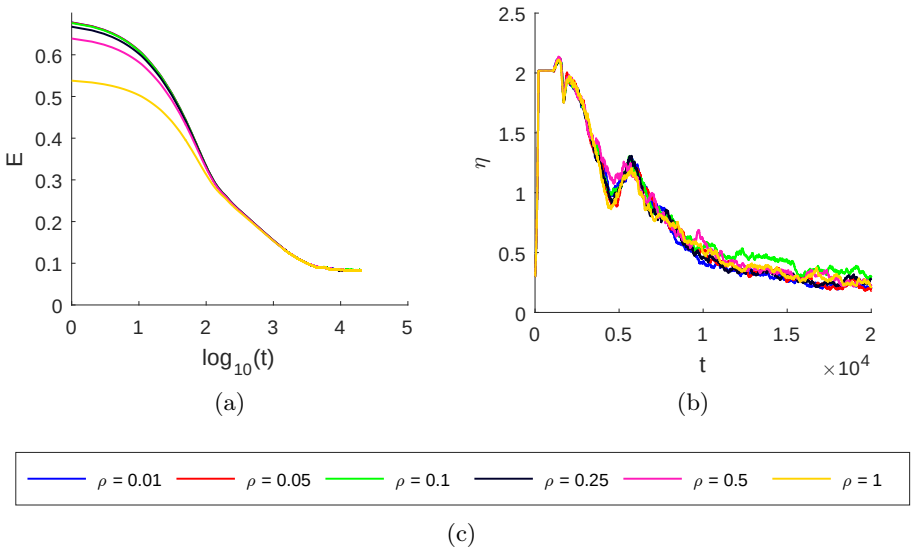
Another approach is Koopmans's method that generalized the LS estimation scheme by developing a maximum likelihood (ML) estimation. In order to assume extended observation vectors and LS it computes an algebraic fit where the norm vector of the ten parameters describing the ellipsoid algebraically (2.7) is equal to 1.

11.2.2 Geometric fitting methods

Some methods have a geometrical definition of error distance, so that the shortest distance from the data point to the surface is computed. Though those geometric methods give better solutions, they require more time for evaluation.

In literature until 2002, Calafiore (Calafiore, 2002) first proposed a feasible approach for multidimensional ellipsoid specific fitting using semidefinite programming (SDP), which minimizes the 2-norm of the algebraic residual vector, who presented the solution of spherical and ellipsoidal fitting based on a "difference-of-squares" (DOS) geometric error criterion for points in n-dimensional space using the equation (11.1) expressed in 3D form:

$$f(\mathbf{c}, \mathbf{A}, r) = \sum \left(d^2(\mathbf{x}_i, \mathbf{c}) - d^2(\tilde{\mathbf{x}}_i, \mathbf{c}) \right)^2 \tag{11.1}$$



(a) (b) (c)

Figure 11.1: Results of our algorithm with perturbed initial estimations of the ellipsoid parameters, generated by the addition of Gaussian noise with standard deviation ρ: (a) error measure E versus time step t, (b) learning rate η versus time step t, (c) legend.

where c is the center of ellipsoid, $\mathbf{A}$ is a symmetric positive definite matrix that defines the shape of the ellipsoid, r is represented by $(\mathbf{x}-\mathbf{c})^T \mathbf{A} (\mathbf{x}-\mathbf{c})$, $d(\mathbf{a}, \mathbf{b})$ denotes a metric distance between the points $\mathbf{a}$ and $\mathbf{b}$, and $\tilde{\mathbf{x}}_i$ is the point on the ellipsoid that lies on the ray $\mathbf{x}_i\mathbf{c}$.

More recent approaches include the Distance Method (Hunyadi and Vajk, 2014) which intends to calculate the orthogonal distance of the points projected onto an ellipsoid. Eberly (Eberly, 2015) defines the error as the normal distance from the point of the cloud to the ellipsoid. The closest point on the ellipsoid is found by a Newton's iteration scheme. He develops methods of hyperplanar fitting and surface fitting for several kinds of quadric curves and surfaces. In the problem of ellipsoidal fitting he focuses on axis-aligned ellipsoids.

11.3 Methodology

Our ellipsoid fitting methodology is presented next. In general terms, it is better to consider criteria based on the sum of the absolute error values to detect the presence of outliers (Seheult et al., 1989; Meer et al., 1991; Huber, 1998; Trucco and Verri, 1998). In this work, we consider a criterion based on the mean absolute deviation from the level set which defines the ellipsoid

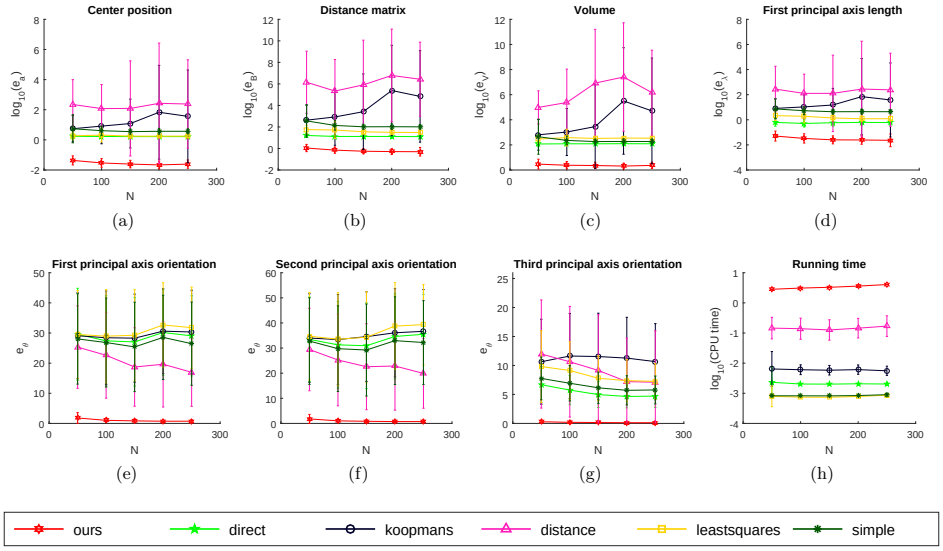


Figure 11.2: Comparison of the performance metrics of all methods varying the number of data points N. Mean and standard deviations of 100 runs are displayed: (a) center error, (b) distance matrix error, (c) volume error, (d) first principal axis length error, (e) first principal axis orientation error (degrees), (f) second principal axis orientation error (degrees), (g) third principal axis orientation error (degrees), (h) running time (seconds).

surface. The proposed algorithm provides good results for a set of points that are not truly geometrically precise ellipsoids but rough ellipsoid or egg-like shapes, e.g., real image applications for agricultural images and many other nearly round shapes.

We start with a review of some basic facts about ellipsoids (Subsection 11.3.1). Then the proposed fitting method and its convergence control procedure are stated in Subsections 11.3.2 and 11.3.3. After that, the initialization procedure is detailed in Subsection 11.3.4 and a summary of the algorithm is provided in Subsection 11.3.5. Finally, a discussion of the most important features of our proposal is given in Subsection 11.3.6.

11.3.1 Preliminaries

The equation of an ellipsoid reads as follows:

$$(\mathbf{x}-\mathbf{a})^T \mathbf{B}^{-1} (\mathbf{x}-\mathbf{a}) = 1 \tag{11.2}$$

where $\mathbf{a} \in \mathbb{R}^3$ is the center of the ellipsoid (a 3-dimensional vector), and $\mathbf{B} \in \mathbb{R}^{3\times 3}$ is a 3×3 positive definite matrix. The eigenvectors of $\mathbf{B}$ are the

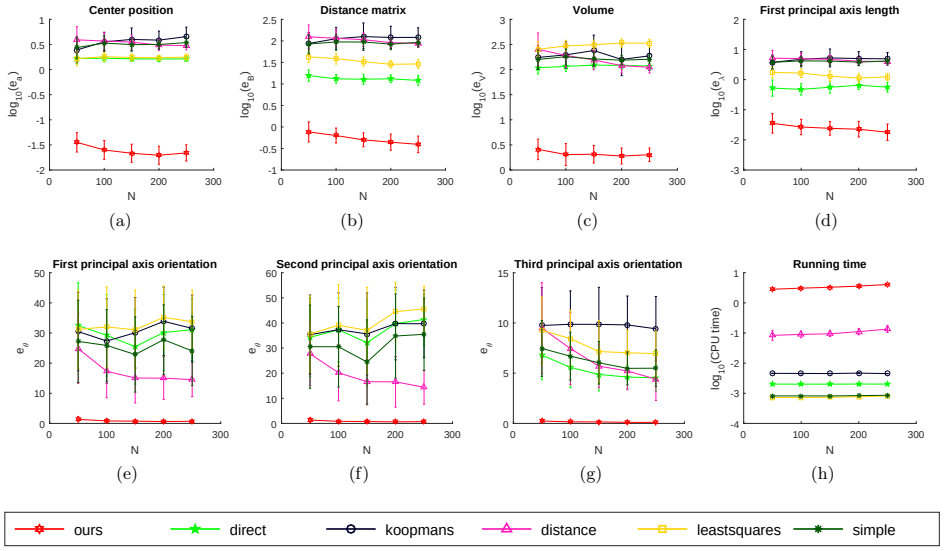


Figure 11.3: Comparison of the metrics of all methods depending on the number of data points. Median and median absolute deviations of 100 runs are displayed: (a) center error, (b) distance matrix error, (c) volume error, (d) first principal axis length error, (e) first principal axis orientation error (degrees), (f) second principal axis orientation error (degrees), (g) third principal axis orientation error (degrees), (h) running time (seconds).

major axes of the ellipsoid, while their associated eigenvalues are the squares of the lengths of these major axes. It must be noted that the ellipsoid surface is the unit level set $f(\mathbf{x}) = 1$ of this function:

$$f(\mathbf{x}) = \sqrt{(\mathbf{x}-\mathbf{a})^T \mathbf{B}^{-1} (\mathbf{x}-\mathbf{a})} \tag{11.3}$$

The ellipsoid can be obtained by affine transformation from the unit sphere with center at the coordinate origin, whose equation is:

$$\mathbf{y}^T\mathbf{y} = 1 \tag{11.4}$$

This is achieved by application of the following affine transform:

$$\mathbf{x} = \mathbf{L}\mathbf{y} + \mathbf{a} \tag{11.5}$$

where $\mathbf{L}$ is a 3×3 lower triangular matrix obtained by Cholesky decomposition of $\mathbf{B}$; please note that $\mathbf{B}$ admits such decomposition because it is positive definite:

$$\mathbf{B} = \mathbf{L}\mathbf{L}^T \tag{11.6}$$

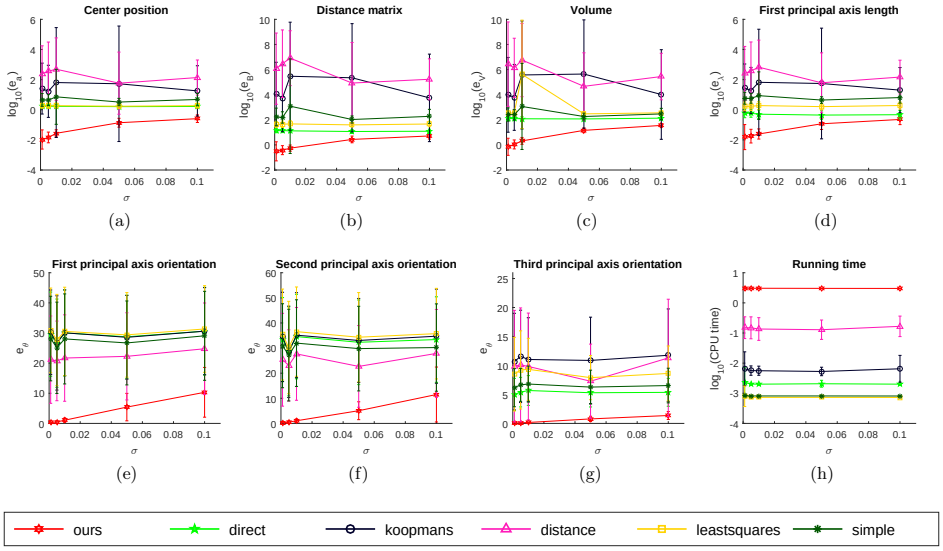


Figure 11.4: Comparison of the performance metrics of all methods depending on the Gaussian noise level for the inliers. Mean and standard deviations of 100 runs are displayed: (a) center error, (b) distance matrix error, (c) volume error, (d) first principal axis length error, (e) first principal axis orientation error (degrees), (f) second principal axis orientation error (degrees), (g) third principal axis orientation error (degrees), (h) running time (seconds).

In order to see this, from (11.5) we have:

$$\mathbf{y} = \mathbf{L}^{-1}(\mathbf{x} - \mathbf{a}) \tag{11.7}$$

Then we substitute (11.7) into (11.4), which leads to (11.2) by application of (11.6).

The ellipsoid divides the space $\mathbb{R}^3$ into two regions. The interior of the ellipsoid is:

$$I = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid (\mathbf{x} - \mathbf{a})^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{a}) < 1 \right\} \tag{11.8}$$

On the other hand, the exterior of the ellipsoid is:

$$O = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid (\mathbf{x} - \mathbf{a})^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{a}) > 1 \right\} \tag{11.9}$$

Next these concepts are applied to robust fitting.

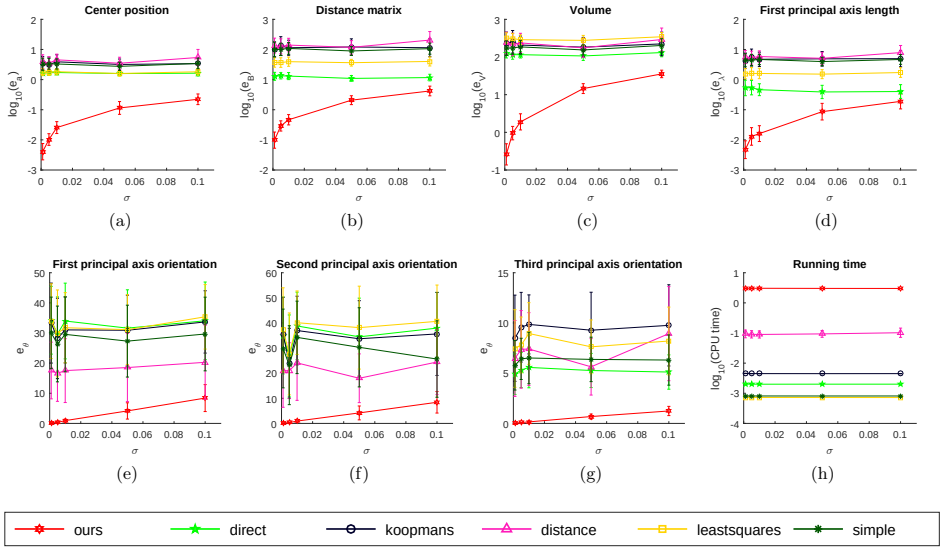


Figure 11.5: Comparison of the metrics of all methods depending on the Gaussian noise level for the inliers. Median and median absolute deviations of 100 runs are displayed: (a) center error, (b) distance matrix error, (c) volume error, (d) first principal axis length error, (e) first principal axis orientation error (degrees), (f) second principal axis orientation error (degrees), (g) third principal axis orientation error (degrees), (h) running time (seconds).

11.3.2 Robust fitting

Let us consider a finite set of M input data points in $\mathbb{R}^3$:

$$S = \left\{ \mathbf{x}_i \in \mathbb{R}^3 \mid i \in \{1, ..., M\} \right\} \tag{11.10}$$

Please note that no normal vectors are provided as inputs, only the data points. The task is to fit an ellipsoid to the input points in S.

Here we propose to use the least absolute error rather than the least squared error, so as to obtain a robust performance criterion (Branham, 1982). Our proposed error measure is:

$$E = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left(\mathbf{x}_i \in O\right) \left(\sqrt{\left(\mathbf{x}_i - \mathbf{a}\right)^T \mathbf{B}^{-1} \left(\mathbf{x}_i - \mathbf{a}\right)} - 1 \right) +$$

$$\frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left(\mathbf{x}_i \in I\right) \left(1 - \sqrt{\left(\mathbf{x}_i - \mathbf{a}\right)^T \mathbf{B}^{-1} \left(\mathbf{x}_i - \mathbf{a}\right)} \right) + \lambda \left\| \mathbf{a} \right\| \tag{11.11}$$

where λ is a positive weighting parameter for the last term which ensures

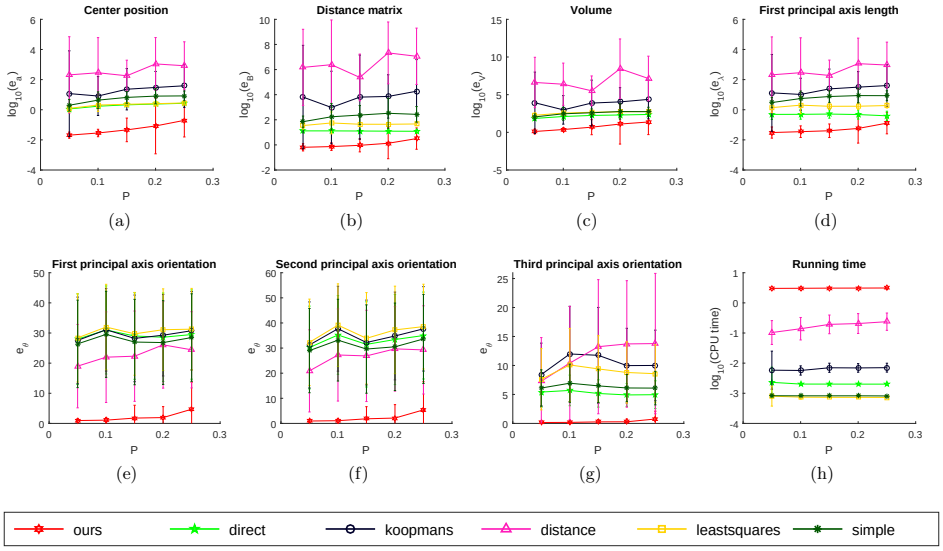


Figure 11.6: Comparison of the performance metrics of all methods depending on the percentage of outliers. Mean and standard deviations of 100 runs are displayed: (a) center error, (b) distance matrix error, (c) volume error, (d) first principal axis length error, (e) first principal axis orientation error (degrees), (f) second principal axis orientation error (degrees), (g) third principal axis orientation error (degrees), (h) running time (seconds).

that the solution does not go to infinity, i.e. $\|\mathbf{a}\| \to \infty$, and $\mathbb{I}$ is the indicator function:

$$\mathbb{I}\left(condition\right) = \begin{cases} 1 & \text{iff } condition \text{ is true} \\ 0 & \text{iff } condition \text{ is false} \end{cases} \tag{11.12}$$

It must be noted that a ellipsoid with a far enough center $\mathbf{a}$ can fit any finite set of points, hence the need for the last term of (11.11). Also, it must be highlighted that the definitions of the interior set (11.8) and the exterior set (11.9) are referred to the current estimation of the ellipsoid parameters $\mathbf{B}$ and $\mathbf{a}$, so both sets are updated as the estimation changes. This implies that all the terms of (11.11) are non negative, so that $E \geq 0$. In turn this means that (11.11) is equivalent to:

$$E = \frac{1}{M}\sum_{i=1}^{M}\left|\sqrt{\left(\mathbf{x}_i - \mathbf{a}\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_i - \mathbf{a}\right)} - 1\right| + \lambda\left\|\mathbf{a}\right\| \tag{11.13}$$

which expresses E as the mean of the absolute errors plus the weighted normalization term.

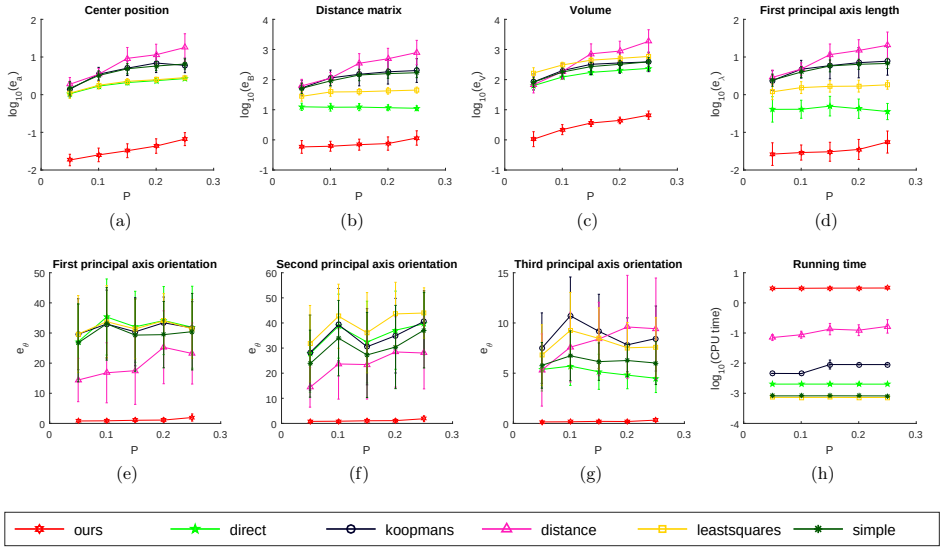


Figure 11.7: Comparison of the metrics of all methods depending on the percentage of outliers. Median and median absolute deviations of 100 runs are displayed: (a) center error, (b) distance matrix error, (c) volume error, (d) first principal axis length error, (e) first principal axis orientation error (degrees), (f) second principal axis orientation error (degrees), (g) third principal axis orientation error (degrees), (h) running time (seconds).

Gradient descent is used to minimize E. Consequently, the gradient with respect to $\mathbf{B}$ is computed:

$$\frac{\partial E}{\partial \mathbf{B}} =$$

$$-\frac{1}{M}\sum_{i=1}^{M}\mathbb{I}\left(\mathbf{x}_i \in O\right)\frac{1}{2}\left(\left(\mathbf{x}_i-\mathbf{a}\right)^T\mathbf{B}^{-1}\left(\mathbf{x}_i-\mathbf{a}\right)\right)^{-\frac{1}{2}}\times$$

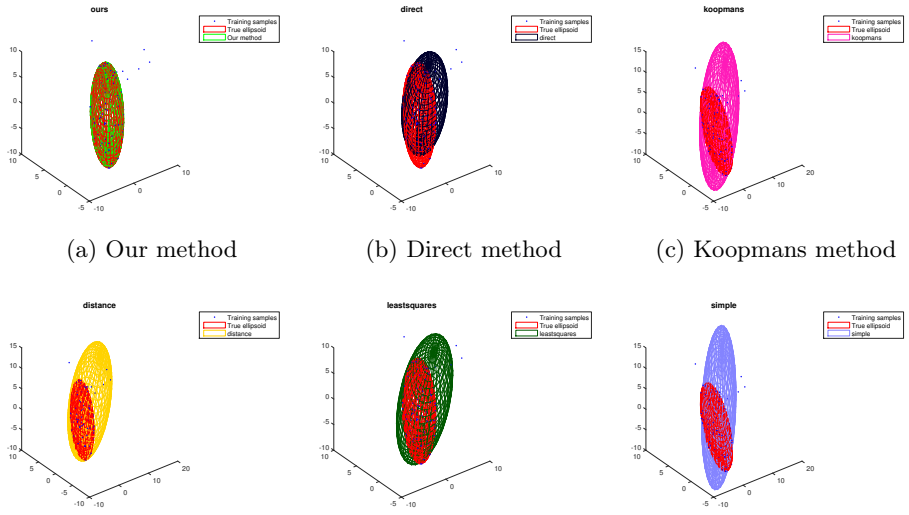
$$\mathbf{B}^{-T}\left(\mathbf{x}_i-\mathbf{a}\right)\left(\mathbf{x}_i-\mathbf{a}\right)^T\mathbf{B}^{-T}$$

$$+\frac{1}{M}\sum_{i=1}^{M}\mathbb{I}\left(\mathbf{x}_i \in I\right)\frac{1}{2}\left(\left(\mathbf{x}_i-\mathbf{a}\right)^T\mathbf{B}^{-1}\left(\mathbf{x}_i-\mathbf{a}\right)\right)^{-\frac{1}{2}}\times$$

$$\mathbf{B}^{-T}\left(\mathbf{x}_i-\mathbf{a}\right)\left(\mathbf{x}_i-\mathbf{a}\right)^T\mathbf{B}^{-T} \tag{11.14}$$

where we note:

$$\mathbf{B}^{-T} = \left(\mathbf{B}^{-1}\right)^T \tag{11.15}$$



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

Figure 11.8: Solutions for synthetic data with outliers.

Next the gradient with respect to $\mathbf{a}$ is obtained as follows:

$$\frac{\partial E}{\partial \mathbf{a}} =$$

$$\lambda \frac{\mathbf{a}}{\|\mathbf{a}\|} - \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}\left(\mathbf{x}_i \in O\right)\left(\left(\mathbf{x}_i - \mathbf{a}\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_i - \mathbf{a}\right)\right)^{-\frac{1}{2}} \times$$

$$\mathbf{B}^{-1}\left(\mathbf{x}_i - \mathbf{a}\right) +$$

$$\frac{1}{M}\sum_{i=1}^{M} \mathbb{I}\left(\mathbf{x}_i \in I\right)\left(\left(\mathbf{x}_i - \mathbf{a}\right)^T \mathbf{B}^{-1}\left(\mathbf{x}_i - \mathbf{a}\right)\right)^{-\frac{1}{2}} \times$$

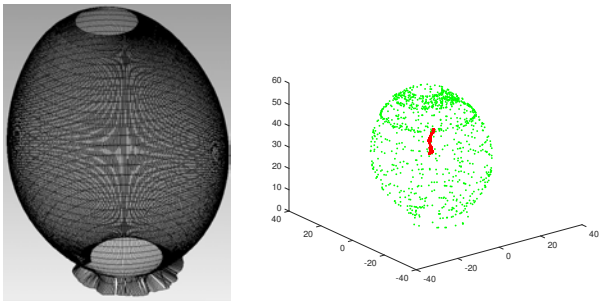
$$\mathbf{B}^{-1}\left(\mathbf{x}_i - \mathbf{a}\right) \tag{11.16}$$

Consequently, the gradient method yields the following update equations:

$$\mathbf{B}\left(t+1\right) = \mathbf{B}\left(t\right) - \eta\left(t\right)\frac{\partial E}{\partial \mathbf{B}}\left(t\right) \tag{11.17}$$

$$\mathbf{a}\left(t+1\right) = \mathbf{a}\left(t\right) - \eta\left(t\right)\frac{\partial E}{\partial \mathbf{a}}\left(t\right) \tag{11.18}$$

where t is the time instant and $\eta\left(t\right)$ is the step size.



(a) (b)

Figure 11.9: Real scanned object with few outliers: (a) 3D scanned points, (b) evolution of the center as estimated by our method.

11.3.3 Robust convergence control

In order to achieve a faster convergence, an adaptive step size $\eta(t)$ is used which depends on the time instant t. The step size should increase when the error decreases in order to advance faster to the local minimum, while it should decrease when the error increases in order to avoid further degradation of the current solution due to wrong updates (Mathews and Xie, 1993; Van Den Doel and Ascher, 2012; Muñoz-Pérez et al., 2014). A potential source of instability in the fitting algorithm is that the error could exhibit a large variability within a given region of the search space. This could lead to wrong updates of the step size, which in turn would increase the variability of the error. Therefore the stability of the method can be enhanced by computing a robust estimation of the current error.

In this work we propose to estimate the error trend in a robust way. To this end, at time instant t the median of the errors at time instants $t, t-1, ..., t-T$ is compared to the median of the errors at time instants $t-T-1, t-T-2, ..., t-2T$, and a binary variable $\xi(t) \in \{-1, 1\}$ is obtained:

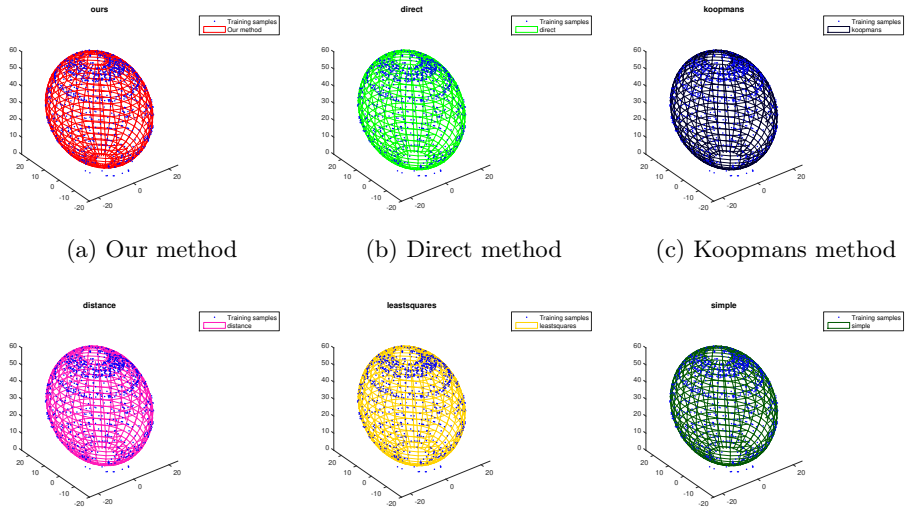
$$\hat{E}_{new}(t) = \text{median}\left(E(t), E(t-1), ..., E(t-T)\right) \tag{11.19}$$

$$\hat{E}_{old}(t) =$$

$$\text{median}\left(E(t-T-1), E(t-T-2), ..., E(t-2T)\right) \tag{11.20}$$

$$\xi(t) = \begin{cases} -1 & \text{iff } \hat{E}_{new}(t) < \hat{E}_{old}(t) \\ 1 & \text{iff } \hat{E}_{new}(t) \geq \hat{E}_{old}(t) \end{cases} \tag{11.21}$$

where $\xi(t) = -1$ indicates that the error exhibits a decreasing trend.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

Figure 11.10: Solutions for scanned real object with few outliers.

Then the step size is updated according to the trend $\xi(t)$:

$$\eta(t+1) = \begin{cases} \rho_{up}\eta(t) & \text{iff } \xi(t) = -1 \text{ and } \eta(t) < \eta_{max} \\ \rho_{down}\eta(t) & \text{iff } \xi(t) = 1 \text{ and } \eta(t) > \eta_{min} \end{cases} \tag{11.22}$$

$$0 < \rho_{down} < 1 \tag{11.23}$$

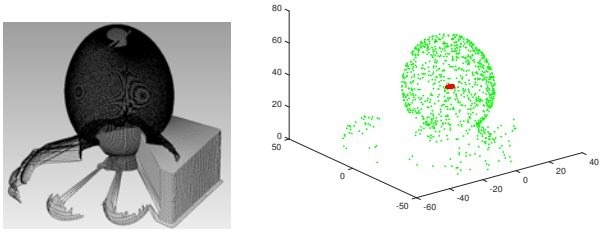
$$\rho_{up} > 1 \tag{11.24}$$

$$\eta_{min} < \eta_{max} \tag{11.25}$$

where ρ_{up} and ρ_{down} control the size of the update of η, and η_{min} and η_{max} set limits for such update. It has been found in practice that by taking $T = 5$, $\rho_{up} = 1.1$ and $\rho_{down} = 0.9$ the step size adaptation procedure is stabilized, so these values have been used in all the reported experiments. In order to save computation time and improve stability, the above step size update procedure is only carried out when t is a multiple of $2T$.

11.3.4 Initialization

Here the initialization procedure to obtain the starting values $\mathbf{B}(0)$ and $\mathbf{a}(0)$ for $t = 0$ is described. It is necessary to produce an initial value of $\mathbf{B}(0)$ which is a positive definite matrix, as indicated in Subsection 11.3.1. To



(a) (b)

Figure 11.11: First real scanned object with many outlier points: (a) 3D scanned points, (b) evolution of the center as estimated by our method.

this end, we take $\mathbf{a}(0)$ as the mean of the training data, and $\mathbf{B}(0)$ as the covariance matrix of the data:

$$\mathbf{a}(0) = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \tag{11.26}$$

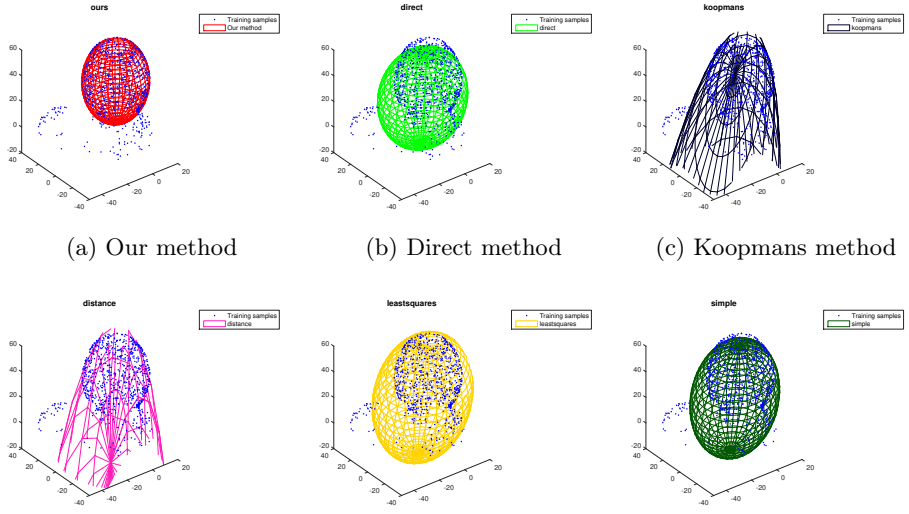
$$\mathbf{B}(0) = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \mathbf{a}(0))(\mathbf{x}_i - \mathbf{a}(0))^T \tag{11.27}$$

Equations (11.26) and (11.27) ensure that $\mathbf{B}(0)$ is positive definite except in case that a degenerate training set is provided which lies in a plane. In case that a more robust initialization is necessary, robust covariance matrix estimation methods are available (Chenouri et al., 2015; Yu et al., 2015; He et al., 2011). They have not been used here because the above procedure yields good results with a reduced computational complexity.

11.3.5 Summary

The proposed algorithm can be summarized as follows:

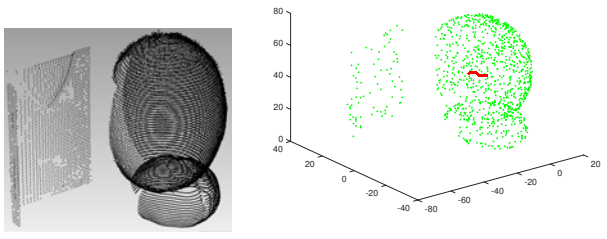
1. Set the initial values of $\mathbf{a}$ and $\mathbf{B}$ according to (11.26) and (11.27), respectively.
2. Record the value of the error E for the current solution.
3. Classify the training data into interior or exterior points with (11.8) and (11.9).
4. Find $\frac{\partial E}{\partial \mathbf{B}}$ and $\frac{\partial E}{\partial \mathbf{a}}$ by equations (11.14) and (11.16), respectively.
5. Update the current values of $\mathbf{B}$ and $\mathbf{a}$ according to (11.17) and (11.18), respectively.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

Figure 11.12: Solutions for the first real scanned object with many outliers.



(a) (b)

Figure 11.13: Second real scanned object with many outlier points: (a) 3D scanned points, (b) evolution of the center as estimated by our method.

6. If the time instant t is a multiple of $2T$, then update the step size according to (11.22).

7. If an error goal or an iteration limit have been reached, then go to step 8. Otherwise, go to step 2.

8. Output the best found solution, i.e. the values of $\mathbf{a}$ and $\mathbf{B}$ associated to the minimum recorded value of E, and then halt.

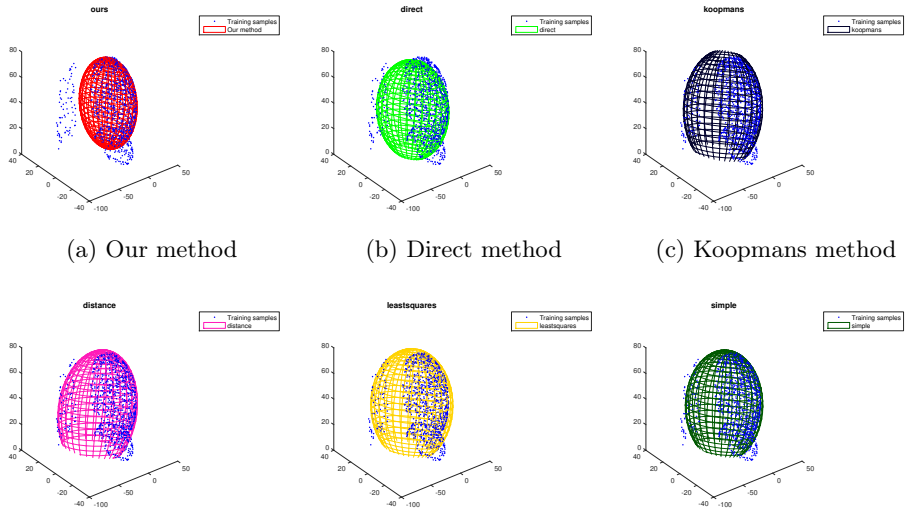
11.3.6 Discussion

In this subsection some important aspects of our proposal are discussed. In particular, the differences of our approach with respect to previous methods are considered, and some directions for future research are outlined.

First of all, it must be studied why standard methods for ℓ_1-norm minimization can not be employed in our approach. One of the best known approaches to replace least squares minimization by a Lp-norm is the Iteratively Reweighted Least Squares (IRLS) method (Daubechies et al., 2010; Welsch, 1977). It employs specially selected weights for the input samples so that Lp-norm minimization can be obtained by solving a weighted least squares problem. It has been successfully applied to various problems in image processing and computer vision (Oliveira et al., 2009; Figueiredo et al., 2007; Comport et al., 2006). However, IRLS is only suitable for problems with a linear objective function. In our case, the proposed objective function (11.13) is nonlinear, which precludes the use of IRLS.

The nonlinearity of the objective function also excludes linear programming as an alternative. Quadratic programming and other more general approaches such as semidefinite programming (Vandenberghe and Boyd, 1996) can not be applied either, because our objective function (11.13) involves the square root of a nonlinear function of the unknown parameters of the ellipsoid $\mathbf{B}$ and $\mathbf{a}$.

Next some possible extensions of our approach are outlined. It must be highlighted that the framework presented here could be extended to other error norms by changing the error measure E in equation (11.11), given a method to compute the gradients $\frac{\partial E}{\partial \mathbf{B}}$ and $\frac{\partial E}{\partial \mathbf{a}}$. It might be applied to fit hyperellipsoids in dimensions higher than 3, since the derivations in Subsections 11.3.1 and 11.3.2 are still valid in $\mathbb{R}^n$ for $n \geq 3$. In principle, it could be extended to other quadratic forms or higher degree polynomials in implicit form $F(\mathbf{x}) = 0$ by defining two regions $F(\mathbf{x}) > 0$ and $F(\mathbf{x}) < 0$. In this case the gradients of $(F(\mathbf{x}))^{\frac{1}{d}}$ and $(-F(\mathbf{x}))^{\frac{1}{d}}$ with respect to the surface parameters would be required, respectively, where d is the degree of the polynomial F.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

Figure 11.14: Solutions for the second real scanned object with many outliers.

11.4 Experimental results

In this section computational experiments for different data sets are reported[1]. The following competing methods have been chosen for comparison:

- Direct Method (DM) (Li and Griffiths, 2004),
- Least Square (LS), (Hunyadi and Vajk, 2014),
- Koopmans method, (Vajk and Hetthessy, 2003),
- Distance, (Hunyadi and Vajk, 2014) and
- Simple (Li and Griffiths, 2004).

Three kinds of data have been employed: synthetic data (Subsection 11.4.1), data from a 3D scanner (Subsection 11.4.2), and finally stereo image pairs from a calibrated stereo camera (Subsection 11.4.3).

11.4.1 Synthetic data

As a first experiment, some synthetic data with added outlier points are used to compare to previous methods. In order to study how our algorithm

[1]The source code and some demos of our approach will be published in http://www.lcc.uma.es/%7Eezeqlr/ellipsoid/ellipsoid.html in case of acceptance

depends on the initial estimation of the ellipsoid parameters, an experiment has been carried out which perturbes the initial estimations of the center $\mathbf{a}$ and the distance matrix $\mathbf{B}$ through the addition of Gaussian noise to the training samples used to compute $\mathbf{a}(0)$ and $\mathbf{B}(0)$, with standard deviations $\rho = 0.01, 0.05, 0.1, 0.25, 0.5, 1$. Please note that the training samples for the subsequent steps of the algorithm are not corrupted by this Gaussian noise.

The error measure E (equation 11.13) and the learning rate η (step size) along the execution steps of the algorithm have been recorded. The results are shown in Figure 11.1. These results demonstrate that regardless of the initial estimation of the ellipsoid, the final measured error is almost the same, even in the case of large initial perturbations. It must be noted that initial values of E are lower for higher values of ρ. This is because higher noise ρ produces a covariance matrix of the data $\mathbf{B}(0)$ with a higher value of $\det(\mathbf{B}(0))$ due to the higher dispersion of the data. This implies that $(\mathbf{x}-\mathbf{a})^T \mathbf{B}^{-1} (\mathbf{x}-\mathbf{a})$ is smaller during the first steps of the algorithm, and hence E is smaller. The learning rate η is not significantly affected by the perturbation in the initial estimation of the ellipsoid, which indicates that the algorithm takes steps of similar size. Hence, the result of the iterative procedure is not degraded under perturbations of the initial ellipsoid parameter estimations.

The second experiment with synthetic data comprises a set of repeated simulation tests to compare the performance of the competing approaches. In order to carry out quantitative comparisons with the competing methods, the center error (11.28), the distance error (11.29), the volume error (11.31), the major axis length error (11.32), and the angular orientation error of the principal axes (11.33) with respect to the true (ground truth) ellipsoid have been computed:

$$\epsilon_{\mathbf{a}} = \|\mathbf{a} - \mathbf{a}_{true}\| \tag{11.28}$$

$$\epsilon_{\mathbf{B}} = \|\mathbf{B} - \mathbf{B}_{true}\| \tag{11.29}$$

$$V = \frac{\pi^{3/2}}{\Gamma\left(\frac{3}{2}+1\right)} \sqrt{\det(\mathbf{B})} \tag{11.30}$$

$$\epsilon_V = |V - V_{true}| \tag{11.31}$$

$$\epsilon_\lambda = |\lambda - \lambda_{true}| \tag{11.32}$$

$$\epsilon_{\theta_i} = \widehat{v_i\, v_{true,i}} \tag{11.33}$$

where the norm $\|\cdot\|$ in (11.28) stands for the Euclidean norm of a vector, the norm $\|\cdot\|$ in (11.29) stands for the 2-norm of a matrix, λ indicates the

Method	$\epsilon_{\mathbf{a}}$	$\epsilon_{\mathbf{B}}$	ϵ_V	ϵ_λ	ϵ_{θ_1}	ϵ_{θ_2}	ϵ_{θ_3}	Runtime (sec)
Ours	0.048975	0.378138	1.683371	0.011273	3.056489	3.058897	0.224226	3.109880
Direct	2.160242	15.301255	136.342838	0.136268	52.008192	56.020176	11.029736	0.103589
Koopmans	3.276190	178.247165	243.922386	6.891759	51.217157	59.683413	15.304854	0.121484
Distance	4.608043	107.567183	434.093471	4.225820	42.076599	63.587081	19.304151	0.448471
Least Square	2.206723	57.779994	289.667286	2.329665	54.819331	58.803372	14.229513	0.007271
Simple	4.208742	230.711858	311.385268	8.521828	53.669883	58.012052	13.494440	0.001149

Table 11.1: Quantitative errors for example synthetic dataset.

principal eigenvalue of $\mathbf{B}$ in (11.32) and v_i is the eigenvector associated to the i-th largest eigenvalue in (11.33).

Three kinds of tests with synthetic data have been considered, by varying the following parameters:

1. The number of inlier training data points randomly drawn from of the ellipsoid surface: $N = 50$, 100, 150, 200, 250.

2. Additive Gaussian noise for the inliers, with $\sigma = 0.001$, 0.005, 0.01, 0.05, 0.1.

3. Percentage of added outliers in the training data: $P = 5$, 10, 15, 20, 25 %.

Please note that the overall number of training data points M is the sum of the inlier and outlier points, i.e. $M = N + \frac{P}{100}N$.

For each kind of test and value of the varying parameter, 100 simulation runs have been carried out to yield significant comparisons. For each simulation run, a new training dataset has been randomly generated and provided as input to all the competing algorithms. The following fixed values were used as one of the parameters was varied: $N = 100$, $\sigma = 0.01$, $P = 10\,\%$.

Three rounds of simulations have been carried out, fixing two of the parameters and varying the other. For each value of each varying parameter and each competing method, we have obtained the mean, median, standard deviation and median absolute deviation over the set of 100 runs. The performance of the competing methods has been evaluated by comparison with the ground truth ellipsoid in terms of the metrics explained before. Also, we obtained the runtime of all methods on a desktop personal computer with a 64-bit quad-core 3GHz CPU and standard hardware. All the methods are implemented as MATLAB scripts, with no use of the GPU.

Figures 11.2 and 11.3 show the comparisons among methods by varying the number of samples N. As seen in Figure 11.2, where the means and the standard deviations of the runs are displayed, the center, distance matrix and volume error are lower than other methods (note that these figures have a logarithmic scale). It is followed by direct, least squares and simple algorithms, whose tendency is the same as ours, i.e. decreasing error as more data samples are employed. However, Koopmans and distance methods

have a different behavior, namely obtaining worse results as N is increased. Values near to zero of the principal axes (length and orientation) confirm that the fitted ellipsoid from our method is the best approximation to the true ellipsoid. In Figure 11.3 the median and median absolute deviations are represented. In both set of graphics it can be observed that our errors are almost always lower than those of the other methods and decrease when more data points are employed, which indicates convergence to the real ellipsoid. The best results in the execution time are achieved by direct and least squares methods, but they are less stable. Our algorithm is not significantly affected in its complexity by random data variations, i.e. its runtime is stable across simulation runs for a given training set size.

Figures 11.4 and 11.5 show the comparisons varying the Gaussian noise level σ for the inlier data points. At low noise levels, our approach has a better behavior than the other methods, because it is the only one which has negative values in the logarithmic scales of the center, distance matrix and volume errors. Our method is less efficient in the orientations, but it is still the best one, attaining errors no larger than 12^{o}. In Figure 11.5 it can be seen that competitors have higher errors even at small values of σ, but when the noise raises, the differences among the methods are smaller. The order of the methods remains the same as when the number of data samples is modified. Furthermore, the running time is not affected by the noise.

Figures 11.6 and 11.7 depict the results of the comparisons varying the percentage of outliers P added to the data set. In general terms, when there are more outliers, worse results are obtained. The precision of our method is the best one, as seen in Figure 11.6. If there are more than 15% of added outliers, our results in the distance matrix and volume errors are close to the direct, leastsquares and simple algorithms. However, the results of the center errors and the axis length and orientation, and especially their variations, allow concluding that our method is better performing. Regarding the median in Figure 11.7 it can be seen that competitors have an error of more than 10^2 in the volume and also the distance matrix with respect to the true ellipsoid. The center and the first principal axis length are also very different. The relative median absolute deviations in the orientations reveal that the rest of the methods are very imprecise when compared with our approach. Again our execution time is slower, around 3 seconds, which is the same as in the other tests.

In order to provide a qualitative assessment of the results with synthetic data, Figure 11.8 depicts an example of the results of the competing methods for a given synthetic dataset. It corresponds to $N = 100$ samples, percentage of added outlier data $P = 10\,\%$, and Gaussian noise level for inliers $\sigma = 0.01$. The associated quantitative performance metrics are shown in Table 11.1. As seen, our method clearly outperforms all its competitors.

	ϵ_V
Ours	67818.428852
Direct	67978.753202
Koopmans	67905.470799
Distance	67825.003952
Least Square	68091.613165
Simple	67971.679099

Table 11.2: Volume errors for the real scanned object with few outlier points.

11.4.2 Scanner data

The second round of experiments considers real objects analyzed by a 3D scanner. The real data used for these experiments have been acquired by using the Roland LPX 1200 3D laser Picza™ scanner with the following characteristics: non contact laser sensor, spot-beam triangulation scanning method, repeat accuracy of ± 0.05 mm and laser wavelength from 645 to 660 nm. It must be highlighted that the quantitative comparisons for real objects can only be done with respect to the volume error (11.31) because the scanner only provides the true volume of the scanned object, while the true center and the true distance matrix are unknown.

Three pieces of meal with near ellipsoidal shapes have been chosen for these experiments, one with few outliers and two with many outliers and significant occlusion. As it is shown in Figure 11.9 we chose an egg as a 3D object with few outlier points. Results are quite similar without any significant differences among the applied methods, as seen in Figure 11.10. In Table 11.2 the volume error among the studied methods is shown. Again, no significant differences are noticed.

Another experiment to compare these methods is based on scanning a 3D object with many outlier points and a significant occluded section. In Figure 11.11, a real object is shown with a dense cloud of outlier points. Figure 11.12 shows the fitting result of the different methods. The robustness of our method is displayed in Figure 11.12a. Table 11.3 reports the comparison of the volume errors using these real data with many outliers. Our approach is shown to consistently outperform its competitors both in qualitative and quantitative terms.

As the last real dataset a scanned piece of fruit with many outlier points and occlusion has been tested (see Figure 11.13) . In Table 11.4 the ellipsoid fitting volume error comparison is shown, which reveals that our method's estimated volume is much closer to the real one than those of the competing approaches. Finally, Figure 11.14 shows the qualitative results. Once again, our method attains a close approximation to the real ellipsoid.

Method	ϵ_V
Ours	65659.271694
Direct	115980.693801
Koopmans	1316316.681929
Distance	2065415.137695
Least Square	166113.326337
Simple	123889.885783

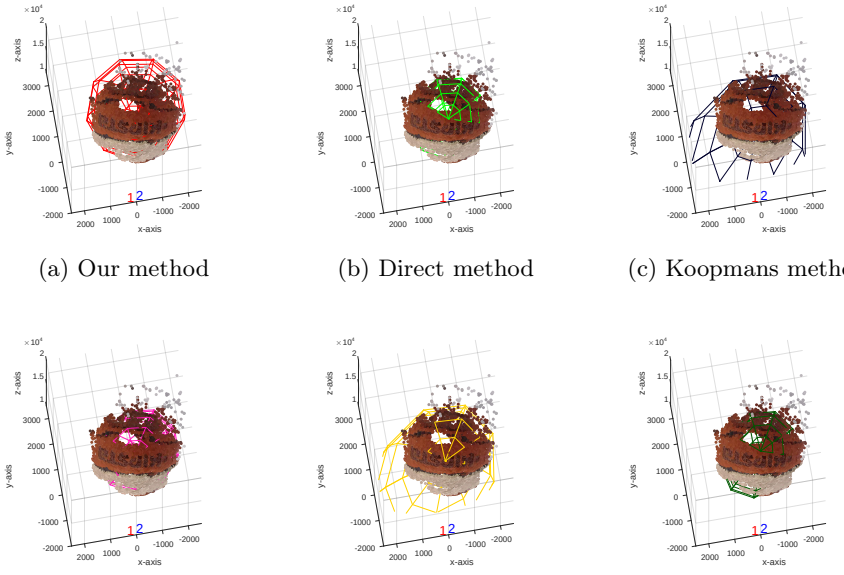
Table 11.3: Volume errors for the first real scanned object with many outlier points.

Method	ϵ_V
Ours	90409.475127
Direct	144565.170593
Koopmans	161886.876864
Distance	171374.993323
Least Square	173355.609190
Simple	154104.187763

Table 11.4: Volume errors for the second real scanned object with many outlier points.

11.4.3 Stereo camera data

The third round of experiments deals with stereo image pairs acquired by a calibrated stereo camera, so that the fitted ellipsoids can be superimposed on the acquired RGB images. These experiments are challenging since about one half of all objects are not visible, i.e. approximately 50% of the object surfaces are missing from the input data sets. The camera employed for these experiments is a JVC GS TD1™. First of all, the camera was calibrated by Zhang's method (Zhang, 2000) to obtain the intrinsic camera parameters. Backprojection errors lower than 0.5 pixels were attained. After that, pairs of images were taken of approximately ellipsoidal objects on a neutral background. The images had 1920×1024 pixel resolution and 8 bit precision for each RGB channel. For each image pair, feature points were searched in the left and right rectified images by the minimum eigenvalue algorithm (Shi and Tomasi, 1994), with minimum quality parameter 0.001 and Gaussian filter size 5×5. Then stereo point pairs were identified by the Kanade-Lucas-Tomasi (KLT) algorithm (Lucas and Kanade, 1981), with the maximum threshold on the forward-backward error set to 1, and 5 pyramid levels. The stereo pairs were located on the intrinsic 3D coordinate system of the camera by triangulation, and finally the resulting 3D data points were provided to the competing ellipsoid fitting algorithms. After the estimated



(a) Our method (b) Direct method (c) Koopmans method

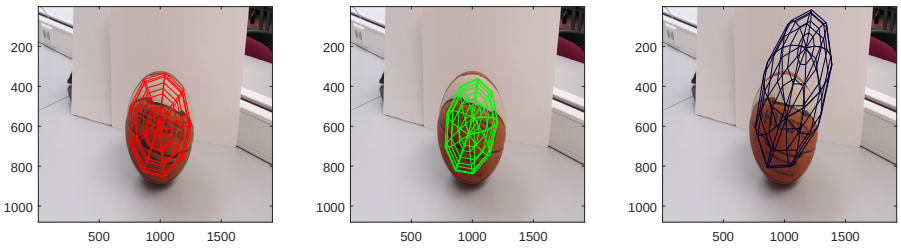
(d) Distance method (e) Least Square method (f) Simple method

Figure 11.15: Solutions for all methods using a 3D stereo reconstruction of a basketball with some outliers. The position of the left lens is marked as '1', and the right lens is marked as '2'.

ellipsoids were computed, they were backprojected to both the left and right images for visual assessment.

Figure 11.15 shows a tridimensional reconstruction of a basketball and the adjustment of each method. It is clear that Koopmans and Least Square methods produce the worst results. Direct, Distance and Simple method obtain an ellipsoid with a similar shape to the basketball, but underestimate the dimensions. This can be better appreciated in Figures 11.16 and 11.17, where the ellipsoids are superimposed on the left and right stereo image pairs. Our fitted ellipsoid covers most of the ball and is centered, which means that our method is less affected by the outliers on the right side of Figure 11.15, while the other methods yield displaced and more stretched ellipsoids.

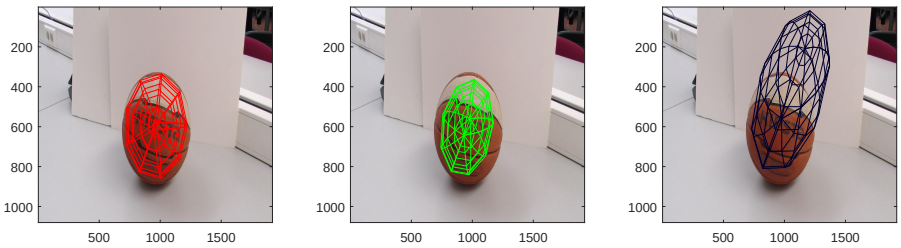
Figure 11.18 shows a tridimensional reconstruction of a loquat and the adjustment of each method. In this case Distance and Least Square methods produce the worst results. Direct, Koopmans and Simple methods obtain almost the same fitted ellipsoid, but our method yields a better estimation of the orientation. Figures 11.19 and 11.20 depict the superimposition on the left and right stereo image pairs. There it can be seen that our method produces an ellipsoid with the same orientation as the loquat, while the others are slightly displaced to the right due to the presence of outliers.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

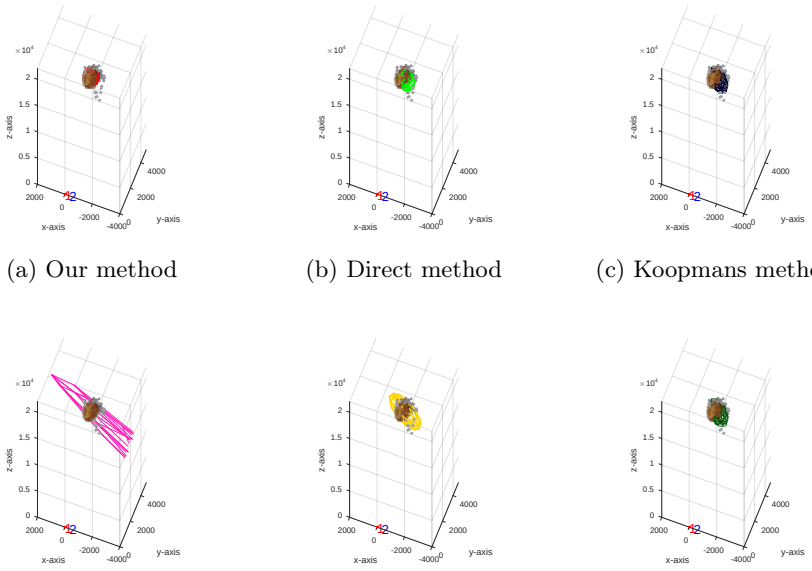
Figure 11.16: Projections on the left image of basketball of the solutions for all methods.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

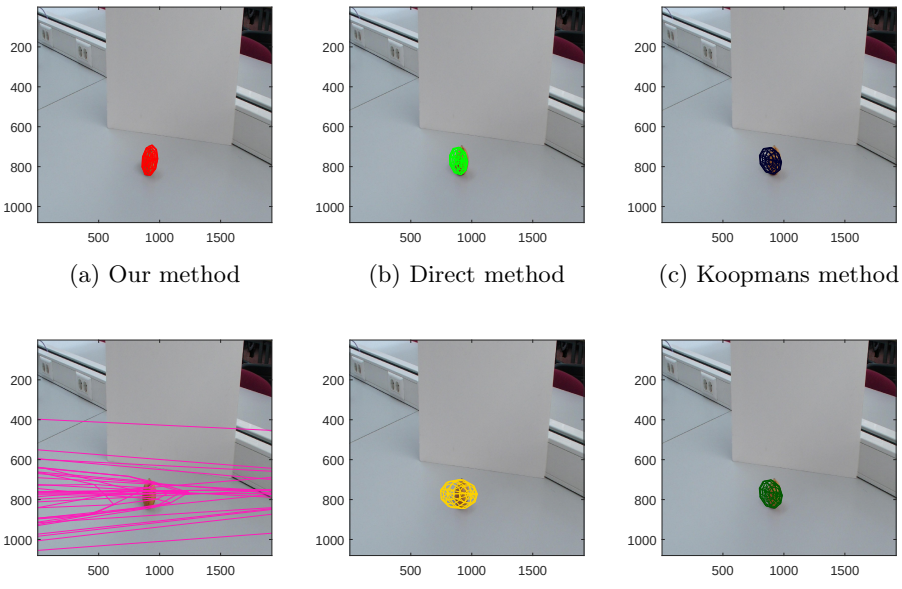
Figure 11.17: Projections on the right image of basketball of the solutions for all methods.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

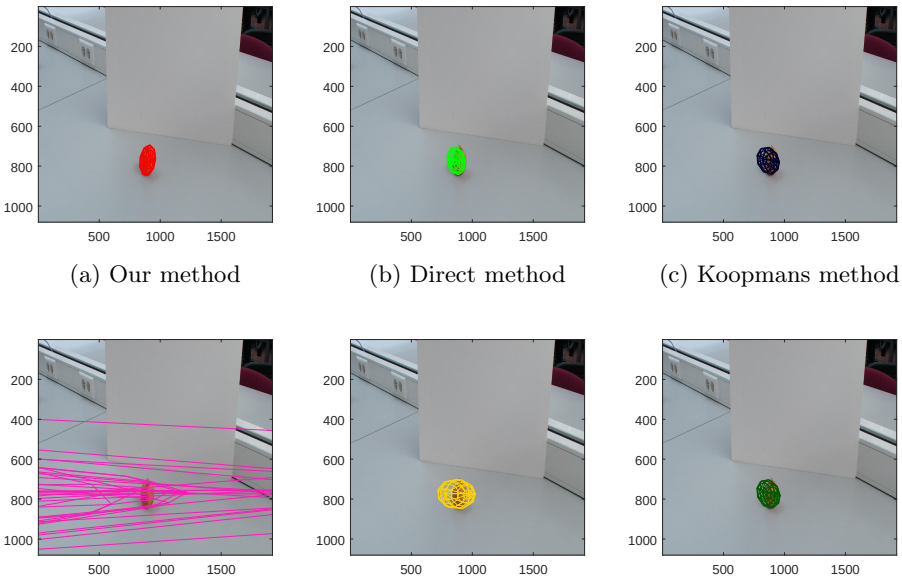
Figure 11.18: Solutions for all methods using a 3D stereo reconstruction of a loquat with some outliers. The position of the left lens is marked as '1', and the right lens is marked as '2'.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

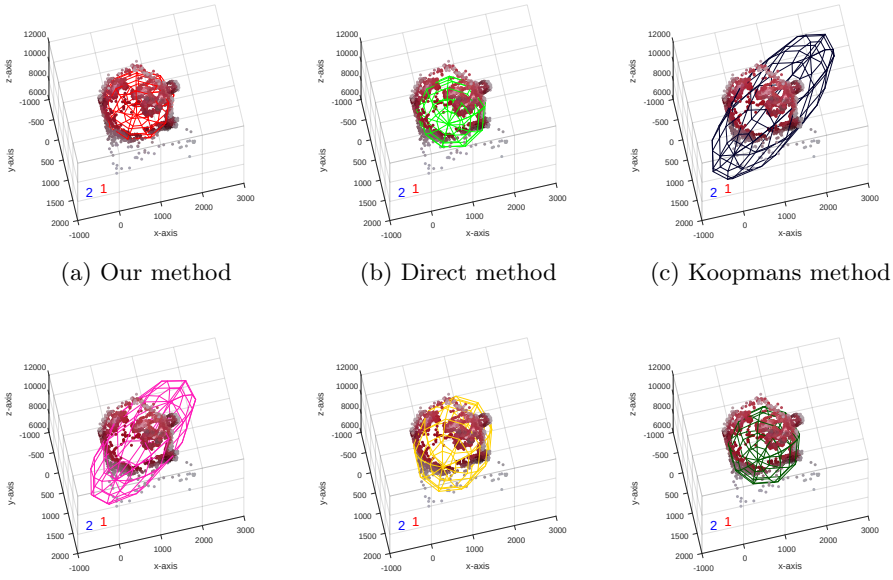
Figure 11.19: Projections on the left image of a loquat of the solutions for all methods.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

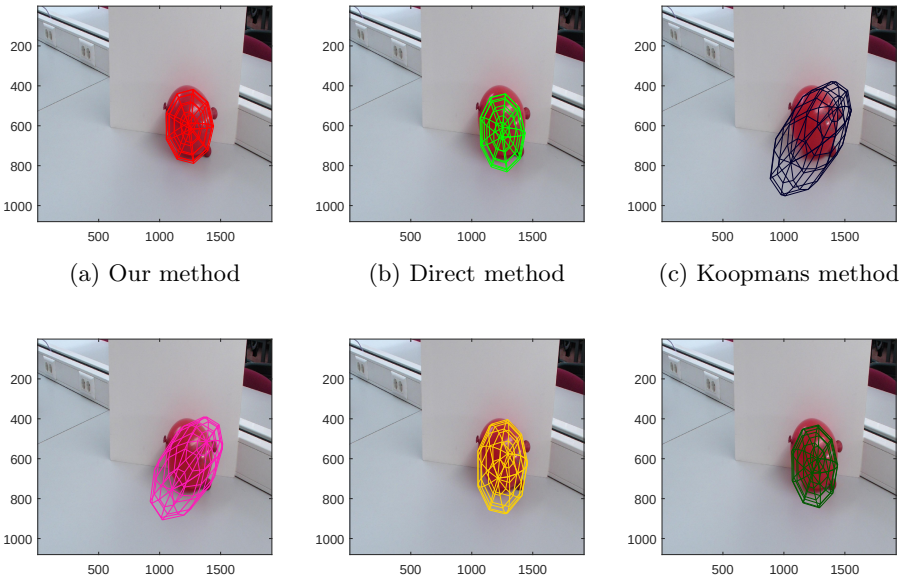
Figure 11.20: Projections on the right image of a loquat of the solutions for all methods.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

Figure 11.21: Solutions for all methods using a 3D reconstruction of a balloon with some outliers. The position of the left lens is marked as '1', and the right lens is marked as '2'.



(a) Our method (b) Direct method (c) Koopmans method

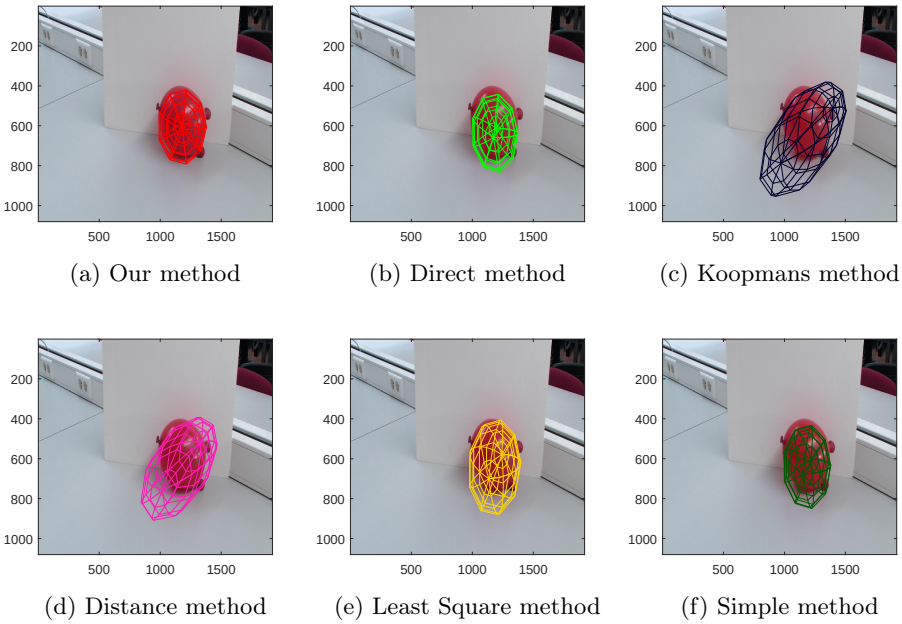
(d) Distance method (e) Least Square method (f) Simple method

Figure 11.22: Projections on the left image of a balloon of the solutions for all methods.

Figure 11.21 shows a tridimensional reconstruction of a balloon and the adjustment of each method. Koopmans and Distance algorithms obtain larger ellipsoids, like the Least Squares method. This might be caused by the irregular shape of the balloon, which has two protuberances on the right side. Direct and Simple methods obtain similar results, but they are affected by the presence of outliers (lower part of the image), which causes a displacement of the estimated ellipsoid. However, this does not happen with our algorithm. Figures 11.22 and 11.23 show the superimposition on the left and right stereo image pairs. It can be seen that our method produces the best fit. Direct, Least Squares and Simple yield a displaced center, while the other two do not achieve a good estimate.

11.5 Conclusions

In this work, we have developed an algorithm to fit an ellipsoid to a set of points by minimizing the mean absolute deviation from the level set which defines the ellipsoid surface. This objective function presents more robust fit results than other compared methods. The proposed algorithm requires an iterative search using a positive definite distance matrix to fit the feature. Nevertheless, independently of the chosen initial parameters the proposed algorithm converges to a finite solution.



(a) Our method (b) Direct method (c) Koopmans method

(d) Distance method (e) Least Square method (f) Simple method

Figure 11.23: Projections on the right image of a balloon of the solutions for all methods.

Algebraic criteria lead to efficient algorithms to the fitting problem but their accuracy can be unacceptable in the presence of outliers. On the other hand, traditional geometric criteria that are used in fitting an ellipsoid are based on a differentiable objective function. They give fast iterative algorithms based on gradients or second derivatives, using nonlinear programming methods. However, these criteria could be very sensitive to outliers or data noise.

Experimental results show that our algorithm is robust since the solutions have a low sensitivity to outliers and occlusion. The robustness of the proposed algorithm is due to the usage of absolute rather than squared errors, since the sum of absolute errors is more robust than the sum of squared errors. Furthermore, the adaptive step control procedure uses the median of the last observed absolute errors; as it is well known the median is a robust estimator. Finally, the solution ellipsoid attains a balance among the outside points and the inside points. These algorithm features are responsible for its robustness against outliers and occlusion. As seen in the simulation tests with synthetic data, our approach consistently outperforms its competitors in all performance metrics, with the exception of the runtime, where our algorithm is slower. Also, a substantially smaller dispersion in the performance results are found for our method, which indicates that our approach is more reliable. The experiments with real data from a 3D scanner and a stereo

camera confirm the results obtained for synthetic data, even for objects with significant occlusion.

Some of the main computational advantages of proposed algorithm are that the objective function usually decreases at each iteration because it uses an adaptive step, the number of iterations is usually reduced, and the operations performed by the algorithm are mainly based on the sum of the absolute errors coming from the distances from each point to the center.

Therefore it can be concluded that a competitive approach for ellipsoid fitting has been proposed, which can be used for datasets with significant amounts of outliers and occlusion. This enhances its applicability to real problems with low quality input data.

Part IV

Super-Resolution in MRI

Chapter 12

Deep learning-based super-resolution of 3D Magnetic Resonance Images by Regularly Spaced Shifting

Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution.

Albert Einstein

Abstract: The image acquisition process in the field of Magnetic Resonance Imaging (MRI) does not always provide high resolution results that may be useful for a clinical analysis. Super-Resolution (SR) techniques manage to increase the image resolution, being especially effective those based on examples that determine a correspondence between patterns of low resolution and high resolution. Deep learning neural networks have been applied in recent years to estimate this association with very competitive results. In this work, the starting point is a convolutional neuronal network to which a regularly spaced shifting mechanism over the input image is applied, with the aim of substantially improving the quality of the resulting image. This hybrid proposal has been compared with several SR techniques using the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and Bhattacharyya Coefficient (BC) metrics. The results obtained on different MR images show a considerable improvement both in the restored image and in the residual image without an excessive increase in computing time.

12.1 Introduction

Super-Resolution is the process whose aim is to generate a High-Resolution (HR) image from a single Low-Resolution (LR) image by means of post-processing techniques. The objective is to improve the image quality taking into account that the solution is not unique (López-Rubio, 2016). Besides its classic applications in the computer vision field, SR is also of great interest for medical imaging. Improvements in SR can be of critical importance, due to the need of obtaining more detailed and realistic images when they are used for medical diagnostic purposes.

In this work, a 3D convolutional neural network previously defined (Pham et al., 2017) (SRCNN3D), is combined with a regularly space shifting mechanism to enhance the quality of the high-resolution image. Thus, the combination of the network output after testing several regularly shifted input images will significantly improve the HR image quality. This technique has been satisfactorily applied to diverse brain MRI image datasets and has been compared with several state-of-the-art SR algorithms. Additionally, another previously published proposal from our research group (Thurnhofer-Hemsi et al., 2018b) has been included in the comparison.

The remaining of the chapter is structured as follows. Section 12.2 describe the proposed model, where in subsection 12.2.1 the convolutional neural network is commented and in subsection 12.2.2 the regularly spaced shifting mechanism is explained. The explanation about how the results are obtained, the different competitive methods and the image datasets analysed, are outlined in Section 12.3 whereas the quantitative and qualitative results are displayed in Section 12.4. To sum up, Section 12.5 discusses the results previously reported and Section 12.6 exposes the conclusions and future works of our approach.

12.2 Theory

In this section deep learning super resolution is carried out on regularly spaced shifted versions of the input image. This technique is proposed to enhance the quality of MR images. In Subsection 12.2.1 the base SRCNN method is outlined. Then in Subsection 12.2.2 the regular shifting algorithm to produce the final estimation of the HR image from the LR one is detailed.

12.2.1 Convolutional neural network

The standard SR reconstruction is carried out through a CNN following these two steps:

1. Given a LR image $\mathbf{X}$, a spline interpolation I is performed in order to obtain a HR image $\mathbf{Z} = I(\mathbf{X})$.

2. A convolutional neural network is applied to restore the image.

The CNN is composed by three blocks of layers. The first two consist in a convolutional layer followed by a Rectified Linear Unit (ReLU). The last layer only performs a convolution, without any ReLU after the filter responses. If we call g_1, g_2, g_3 for each of these operations, the net computes a HR image

$$g = g_3 \circ g_2 \circ g_1 \tag{12.1}$$

and the restoration based on a residual learning technique and computed minimising the Euclidean loss

$$\tilde{f} = arg\, min_g \sum ||\mathbf{Y} - g(\mathbf{Z})||^2 \tag{12.2}$$

where $\mathbf{Y}$ is the HR image corresponding to $\mathbf{X}$.

The first convolutional layer applies 64 filters of size $9 \times 9 \times 9$, 32 of size $1 \times 1 \times 1$ in the second, and one filter of size $5 \times 5 \times 5$ in the last layer. Several overlapping patches per image are extracted from a set of HR reference images in order to have enough samples to train the network, and feature maps are computed. For each patch, a down-sampling and up-sampling is applied and a relationship is established to learn an end-to-end mapping between LR and HR images.

This neural network was used as a base for the proposed model described in the next section. It is called SRCNN3D and specific details about its implementation can be found in (Pham et al., 2017).

12.2.2 Regularly spaced shifting model

Our proposed regular shifting method is presented in this subsection. It consists in combining the outputs produced by the SRCNN network for shifted versions of the original input image, where the shift vectors are varied according to a regular rectangular pattern. The motivation for this approach is that each shifted input image produces a slightly different output when processed by the SRCNN network, so that a number of variants of the output are obtained as different shift vectors are considered. After that, the obtained output images can be averaged in order to produce an ensemble output. This averaging operation smooths out a certain amount of noise, thereby increasing the quality of the final combined output image. As compared to our previous random shifting approach (Thurnhofer-Hemsi et al., 2018b), the regular shifting approach proposed here ensures that the set of considered shift vectors is balanced within the set of all possible shift vectors. The regular shifting approach is based on numerical integration theory, while our previous approach relies on statistical approximation theory.

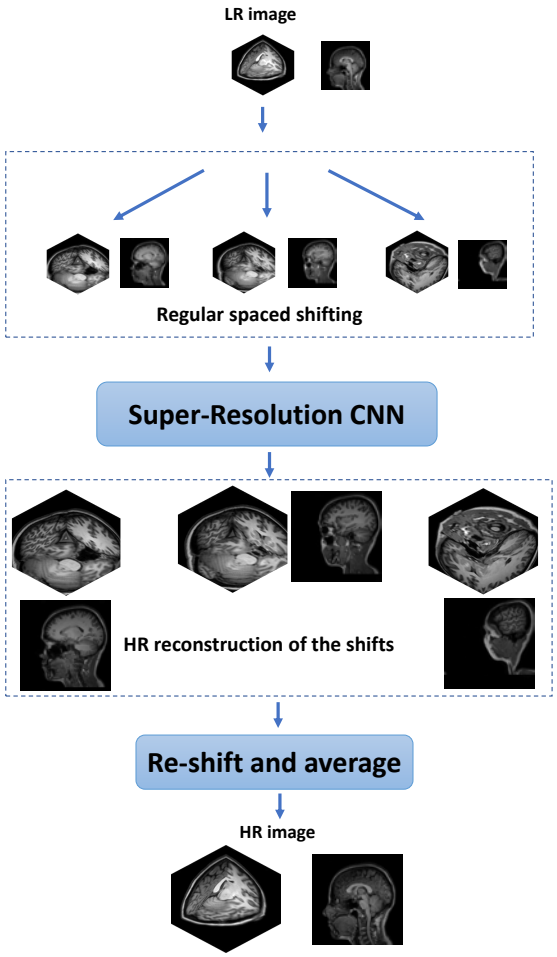


Figure 12.1: Scheme of the proposed algorithm.

Given an LR image $\mathbf{X}$, the SRCNN model learns an approximation $\tilde{f}(\mathbf{X})$ of the underlying HR image $f(\mathbf{X})$:

$$f(\mathbf{X}) = \tilde{f}(\mathbf{X}) + \tilde{\epsilon}(\mathbf{X}) \tag{12.3}$$

where $\tilde{\epsilon}(\mathbf{X})$ is the approximation error. Let us note:

$$\mathcal{V} = [0, Z) \times [0, Z) \times [0, Z) \tag{12.4}$$

where Z is a window size (in voxels). Shifted versions LR image can be considered as inputs to the SRCNN, where $\mathbf{a} \in \mathcal{V}$ is a shift vector. Therefore a different approximation $\tilde{f}_{\mathbf{a}}(\mathbf{X})$ of the underlying HR image f is obtained

for each shift vector $\mathbf{a}$:

$$\bar{f}_{\mathbf{a}}(\mathbf{X}) = \tilde{f}(\mathbf{X} \ll \mathbf{a}) \gg \lambda\mathbf{a} \tag{12.5}$$

$$\bar{\epsilon}_{\mathbf{a}}(\mathbf{X}) = \tilde{\epsilon}(\mathbf{X} \ll \mathbf{a}) \gg \lambda\mathbf{a} \tag{12.6}$$

$$f(\mathbf{X}) = \left(\tilde{f}(\mathbf{X} \ll \mathbf{a}) + \tilde{\epsilon}(\mathbf{X} \ll \mathbf{a})\right) \gg \lambda\mathbf{a} =$$

$$\left(\tilde{f}(\mathbf{X} \ll \mathbf{a}) \gg \lambda\mathbf{a}\right) + (\tilde{\epsilon}(\mathbf{X} \ll \mathbf{a}) \gg \lambda\mathbf{a}) =$$

$$\bar{f}_{\mathbf{a}}(\mathbf{X}) + \bar{\epsilon}_{\mathbf{a}}(\mathbf{X}) \tag{12.7}$$

where $\ll$ and $\gg$ stand for the image left shift and right shift operators, respectively, and λ is the super-resolution zoom factor.

Given an LR image $\mathbf{X}$, let $\varphi_{\mathbf{X}}$ be a constant function which takes the shift vector $\mathbf{a}$ as argument:

$$\varphi_{\mathbf{X}}(\mathbf{a}) = f(\mathbf{X}) \tag{12.8}$$

Here it must be highlighted that the domain of $\varphi_{\mathbf{X}}$ is the set of valid values for $\mathbf{a}$, i.e. $\mathbf{a} \in \mathcal{V}$. Moreover, from (12.8) we can write:

$$f(\mathbf{X}) = \frac{1}{Z^3} \int_{\mathcal{V}} \varphi_{\mathbf{X}}(\mathbf{a})\, d\mathbf{a} \tag{12.9}$$

For each $\mathbf{a}$, from (12.7) and (12.8) we get an approximation for $\varphi_{\mathbf{X}}(\mathbf{a})$:

$$\varphi_{\mathbf{X}}(\mathbf{a}) = \bar{f}_{\mathbf{a}}(\mathbf{X}) + \bar{\epsilon}_{\mathbf{a}}(\mathbf{X}) \tag{12.10}$$

Now the standard approximation (12.3) of the underlying function f by the SRCNN model can be rewritten as follows:

$$f(\mathbf{X}) = \tilde{f}(\mathbf{X}) + \tilde{\epsilon}(\mathbf{X}) = \varphi_{\mathbf{X}}(\mathbf{0}) = \bar{f}_{\mathbf{0}}(\mathbf{X}) + \bar{\epsilon}_{\mathbf{0}}(\mathbf{X}) \tag{12.11}$$

This means that approximating $f(\mathbf{X})$ by $\tilde{f}(\mathbf{X})$, which is what the original SRCNN does, amounts to a zeroth order approximation of $\varphi_{\mathbf{X}}(\mathbf{0})$ obtained by taking a single noisy value $\bar{f}_{\mathbf{0}}(\mathbf{X})$. We propose to estimate $f(\mathbf{X})$ by approximating the integral in (12.9). In turn, the integral is estimated by averaging approximations of $\varphi_{\mathbf{X}}(\mathbf{a})$ given by (12.10) and taken at regularly spaced values of $\mathbf{a}$:

$$\hat{f}(\mathbf{X}) = \frac{1}{M} \sum_{j=1}^{M} \bar{f}_{\mathbf{a}_j}(\mathbf{X}) \tag{12.12}$$

where the $\mathbf{a}_j$ are taken at regularly spaced values from $\mathcal{V}$.

Now, it is reasonable to think that $\hat{f}(\mathbf{X})$ is a better approximation to $f(\mathbf{X})$ than the original approximation $\bar{f}_{\mathbf{0}}(\mathbf{X})$. On one hand, $\hat{f}(\mathbf{X})$ is built as an approximation of an integral from many regularly spaced noisy observations, where the noise function $\bar{\epsilon}_{\mathbf{a}}(\mathbf{X})$ in (12.10) is expected to be smooth and have zero mean with respect to $\mathbf{a}$, i.e. $E_{\mathbf{a}}[\bar{\epsilon}_{\mathbf{a}}(\mathbf{X})] = \mathbf{0}$, because $\bar{f}_{\mathbf{a}}(\mathbf{X})$ is the output of the SRCNN as the input is shifted by $\mathbf{a}$. Therefore, the noise might be partially averaged out by the integral. On the other hand, $\bar{f}_{\mathbf{0}}(\mathbf{X})$ is built as a zeroth order approximation from a single noisy observation. Hence we propose to use $\hat{f}(\mathbf{X})$ as an approximator to $f(\mathbf{X})$. In practice the $\mathbf{a}_j$ are constrained to be integer vectors, so that fractional shifts are not necessary, since fractional shifts would pose a difficult problem themselves.

The proposed algorithm reads as follows:

1. Given M, compute the following set of vectors:

$$\mathcal{S}_M = \{\mathbf{a}_m\}_{m\in\{1,\dots,M\}} \tag{12.13}$$

$$\mathbf{a}_m = (a^{(m)}, a^{(m)}, a^{(m)}) \in \mathbb{N}^3 \tag{12.14}$$

   where $a^{(m)}$ is obtained as:

$$a^{(m)} = (m-1)R \tag{12.15}$$

   with R being a tunable parameter of the algorithm which indicates the spacing between consecutive shifts.

2. For an input LR image $\mathbf{X}$, compute M circularly image shifts:

$$\{\mathbf{X} \ll \mathbf{a}_m,\ \mathbf{a}_m \in \mathcal{S}_M\} \tag{12.16}$$

3. Apply the CNN to obtain a set of HR images:

$$\{\tilde{f}(\mathbf{X} \ll \mathbf{a}_m),\ \mathbf{a}_m \in \mathcal{S}_M\} \tag{12.17}$$

4. Recompose the images considering the super-resolution zoom factor to obtain $\{\bar{f}_{\mathbf{a}_m}(\mathbf{X}),\ \mathbf{a}_m \in \mathcal{S}_M\}$, and then compute the final restored HR image $\bar{f}(\mathbf{X})$ following Eq. (12.12).

A schematic depiction of the operation of our algorithm is shown in Figure 12.1. It must be noted that (12.15) corresponds to the following window size:

$$Z = MR \tag{12.18}$$

12.3 Material and Methods

Description of the experiments we have carried out, and also datasets and algorithms are reported in this section. The LR image generation procedure, the software and hardware that we have used, and the selected performance metrics for comparison between methods are specified in Subsection 12.3.1. Then, the set of MR images are described in 12.3.2. The set of tuned parameters of our proposal and the tuning experiments are explained in Subsection 12.3.3 and the descriptions of the competing algorithms in Subsection 12.3.4.

12.3.1 Methods

Firstly, we need to describe how the LR images are generated since we only have available the HR ones. Input data for each method were obtained following this procedure:

1. HR images were cropped in relation to the zoom factor to be applied, in order to avoid fractional values.
2. A three-dimensional Gaussian filter was applied, with standard deviation equal to 1.
3. Finally, the LR image is generated applying a cubic interpolation through the *imresize3* function of MATLAB and its default parameters.

As the convolutional neural network used for the super-resolution step we selected the SRCNN3D method. It has demonstrated a considerable effectiveness compared with other state-of-art deep learning methods. This is a CNN method where a huge set of reference patches are extracted from each training HR image and compared with the same set of downsampled and upsampled patches in order to update the network weights.

An online available pre-trained model was used for the experiments. This network was trained over 470000 iterations, with 10 images from Kirby dataset (Landman et al., 2011) for training the CNN (images 33-42), using momentum of 0.9, learning rate of 0.0001 and batch size of 256. Stochastic gradient descent was used for model optimization.

All the experiments described in this work were carried out on a 64-bit Personal Computer with an eight-core Intel i7 3.60GHz CPU, NVidia Titan X GPU, 32 GB RAM and standard hardware, using MATLAB R2017b. The deep learning method network has been developed using the Caffe package (Jia et al., 2014) on a Python framework.

Three different quality measures were used to evaluate and compare the proposed method:

- *Peak Signal-to-Noise Ratio (PSNR)*, measured in (decibels) dB, which is commonly used in medical image processing (higher is better).

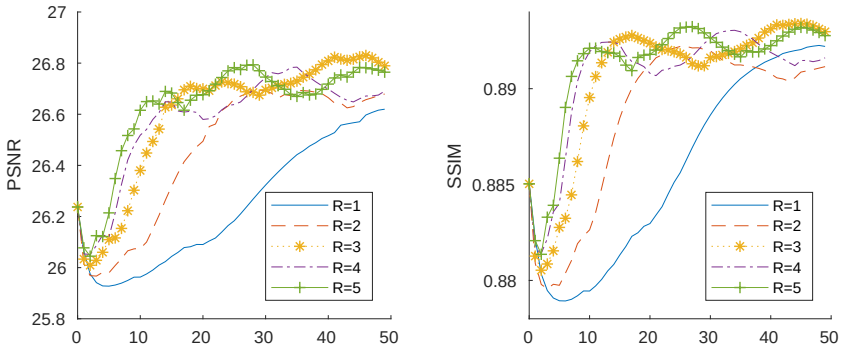


Figure 12.2: Evolution of the PSNR and SSIM metrics (higher is better) when varying the number of shifts employed for the reconstruction. Five different spacing values are plotted. Image 1 of the Kirby 21 dataset, image 80 of disc1 of the OASIS dataset and T1 noiseless image of Brainweb were used for the analysis.

- *Structural Similarity Index Measure (SSIM)* (Wang et al., 2004), which focuses on structural similarities between images (higher is better):

$$SSIM(x,y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \tag{12.19}$$

  where $\mu_x$and μ_y are the mean value of images x and y, σ_x and σ_y are the standard deviation of images x and y, σ_{xy} is the covariance of x and y, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ (default values were used: $L = 1$ is the dynamic range, $k_1 = 0.01$ and $k_2 = 0.03$).

- *Bhattacharyya Coefficient (BC)* (Bhattacharyya, 1946), which measures the closeness of the two discrete pixel probability distributions P and $\hat{P}$ corresponding to the Ground Truth (GT) and modeled images with values in the range $[0, 255]$:

$$BC = \sum_{j=0}^{255} P(j)\hat{P}(j) \tag{12.20}$$

  where $BC \in [0, 1]$ and higher is better.

Also, CPU time was measured for each method. In the case of the methods based on the repetition of CNN testings as many times as shifted images were generated, we measured the processing time sequentially considering only one GPU. Nevertheless, they could be parallelized in different GPUs to improve the total execution time.

From a qualitative point of view, we compared the competing methods using both restored and residual HR images. This residual image r was

Table 12.1: Considered parameter values for our method

Parameter	Value
Spacing R	3
Number of shifts M	47

computed as the difference between the original HR image h and the super-resolved one s:

$$r = h - s \tag{12.21}$$

A darker residual image imply a better performance, as the differences between the original HR image and the restored must be close to zero. Its values were subtracted to the positive constant 0.5 and color maps were adjusted for a better visualization and discrimination between methods.

12.3.2 Datasets

Seven different images obtained from different datasets were considered in order to evaluate the results of the SR algorithms.

- Three T1-weighted MRI images of the Kirby 21 (images 5, 10 and 11) (Landman et al., 2011). These data were acquired using a 3-T MR scanner with a $1.0 \times 1.0 \times 1.2\,mm^3$ voxel resolution over an field-of-view (FOV) of $240 \times 204 \times 256\,mm$ acquired in the sagittal plane. They are named as KKI2009XXMPRAGE.

- Two T1 images of the Open Access Series of Imaging Studies (OASIS) dataset (images 1 and 2 of the cross-sectional data) (Marcus et al., 2007) . Data were acquired on a 1.5-T Vision scanner with a $1.0 \times 1.0 \times 1.25\,mm^3$ voxel resolution over an FOV of $256 \times 256\,mm$. They are named as OAS1_000X_MR1_mpr1_anon.

- One image of the Internet Brain Segmentation Repository (IBSR) public dataset (Worth, 2010). It is named IBSR_07, it has image size $256 \times 256 \times 128$, with $1.5 \times 1.0 \times 1.0\,mm^3$ voxel resolution.

- A T1-weighted image was acquired at the Medical Research Center of the University of Málaga (CIMES)[1] using a 3-T MR scanner with a $0.93 \times 0.93 \times 1.0\,mm^3$ voxel resolution over an FOV of $256 \times 256\,mm$.

All these images are different from the CNN training dataset. Image 5 of Kirby 21 was only analyzed with zoom factor 3.

[1] www.cimes.es

12.3.3 Parameter selection

Our model depends on the tuning of two parameters: the spacing of the shifting R and the number of shifts M. For this necessary prior task, we employed three images different from those selected for the final experiments:

- Image 1 of the Kirby 21 dataset: KKI2009-01-MPRAGE.
- Image 80 of the OASIS dataset: OAS1_0080_MR1_mpr-1_anon.
- A T1-weighted image from the Brainweb[2] simulated database (slice thickness 1 mm, 0 % noise level and $RF = 0$).

The default values of the SRCNN3D model were used, so no tuning was needed. We computed the Peak Signal-to-Noise Ratio (PSNR) and SSIM measures for all these images and we plotted the mean values for each spacing value $R \in \{1, 2, 3, 4, 5\}$ varying the number of shifts M between 0 and 50, where $M = 0$ corresponds to the base method SRCNN3D. The evolution of these metrics displayed in Figure 12.2. Despite the fact that the frequency of the oscillations in the performance measures raises as the number of shifts increases, a stabilization is observed above 30 shifts. The performance values for $R = 3$ seem to be higher than those for other values of R. According to these results, a parameter selection has been carried out. The chosen parameters are reported in Table 12.1.

12.3.4 Competitors

Seven SR algorithms were used to compare our proposal[3], which was named as *SRCNN3D+RegSS*:

- *Spline*: bicubic spline interpolation as implemented in MATLAB (Mathworks Inc.).
- *NLMU* (non-local means upsampling) (Manjón et al., 2010b): recover some of high frequency information by using a data-adaptive patch-based reconstruction in combination with a subsampling coherence constraint.
- *LRTV* (low-rank total variation) (Shi et al., 2015): low-rank regularization and total variation techniques were used to integrate both local and global information for image reconstruction.
- *SRCNN3D* (Pham et al., 2017): three-dimensional convolutional neural network trained with patches of HR brain images. Instead of learning the mapping directly from the LR space to the HR one, it estimates

[2] http://mouldy.bic.mni.mcgill.ca/brainweb/
[3] https://bitbucket.org/icaiuma/srcnn3d_regss/

a mapping from the LR space to the missing high-frequency components.

- *SRReCNN3D-10L* (Pham et al., 2019): A generalization of SRCNN3D where 10 blocks of Convolution+ReLU are used in the network architecture. A pre-trained model available online was used. Only works for zoom factor 2, so experiments with zoom 3 omit this competing method.

- *SRReCNN3D-20L* (Pham et al., 2019): A generalization of SRCNN3D where 20 blocks of Convolution+ReLU are used in the network architecture. A pre-trained model available online was used. Only works for zoom factors 2 and 3.

- *VDSR* (Kim et al., 2016): very deep convolutional network with 20 layers based on residual learning with extremely high learning rates. Contextual information is exploited by cascading small filters.

- *EDSR* (Lim et al., 2017): compact deep network which removes unnecessary modules from conventional ResNet architecture, also based on residual scaling techniques.

- *SRCNN3D+RndS* (Thurnhofer-Hemsi et al., 2018b): A previous approach based on the use of a random shifting technique.

The first three have been implemented in MATLAB. The convolutional network called *SRCNN3D*, which is used by our proposal, and also *SRReCNN3D-10L* and *SRReCNN3D-20L* have been developed using Caffe package (Jia et al., 2014) on a Python framework. *VDSR* was created and integrated in MATLAB. There are versions in Torch and Pytorch of the *EDSR* network, so we used the last one[4]. As both *VDSR* and *EDSR* networks were designed for 2D images, we applied a straightforward strategy combining results from coronal, axial, and sagittal views by computing the average of this three image reconstructions (Chen et al., 2018c; Pham et al., 2017).

12.4 Experimental results

A quantitative evaluation of the performance of the ten different methods using zoom factor 2 is summarized in Figure 12.3. The mean and the standard deviation for each method are shown with horizontal error bars, for the six tested images (all except image 5 of Kirby21), using a super-resolution zoom factor of 2. Traditional methods, i.e. *Spline*, *NLMU* and *LRTV*, perform clearly worse than deep learning methods, both in mean and standard deviation values. We can see that the *LRTV* method outperforms the other

[4] https://github.com/thstkdgus35/EDSR-PyTorch

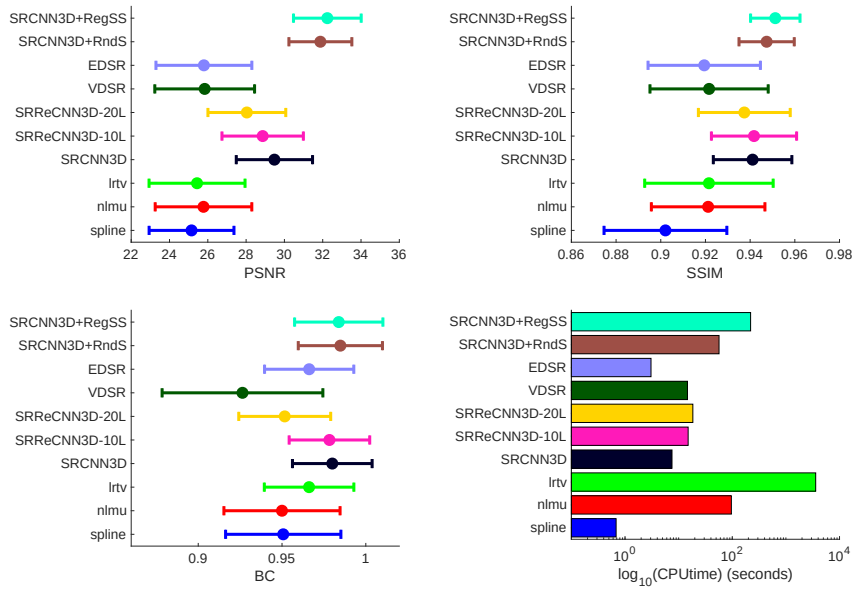


Figure 12.3: Comparison of the PSNR, SSIM, BC (higher is better) and CPU time for the ten methods. Mean and standard deviation of the results for all the test images except image 5 of Kirby21 are displayed, using $\lambda = 2$.

two in BC, but PSNR and SSIM values still remain similar. A second block of methods can be distinguish from the others, composed by the *SRCNN3D* algorithm and its generalisations *SRReCNN3D-10L* and *SRReCNN3D-20L*. An improvement of all the metrics can be appreciated with respect to the conventional methods, reaching 29 dB of PSNR and a SSIM value of 0.94. On the other hand, *VDSR* and *EDSR* networks do not reach good results with respect *SRCNN3D*.

A second amelioration of all the metrics can be seen when comparing the shifting models with the isolated CNN-based methods. PSNR increases from 29 dB up close to 32 dB, and SSIM and BC are also improved, which indicates that the local brain structures are restored in a better way. With the proposed regular spaced shifting model *SRCNN3D+RegSS* PSNR values of 33 dB are achieved. Actually, better values of SSIM are reported reducing the standard errors. BC values remains flat.

Figure 12.3 also shows the mean processing time (in a logarithmic scale) required for the execution of each method. The *Spline* method is the fastest one, followed by *EDSR*. However, LR images are poorly restored as seen before. *NLMU* and *LRTV* need long times to process a single image and are also very dependent on its dimensions. The higher the number of voxels, the more time they use to generate the HR image. Our proposal and *SRCNN3D+RndS* also require time, which depends on the number of shifts.

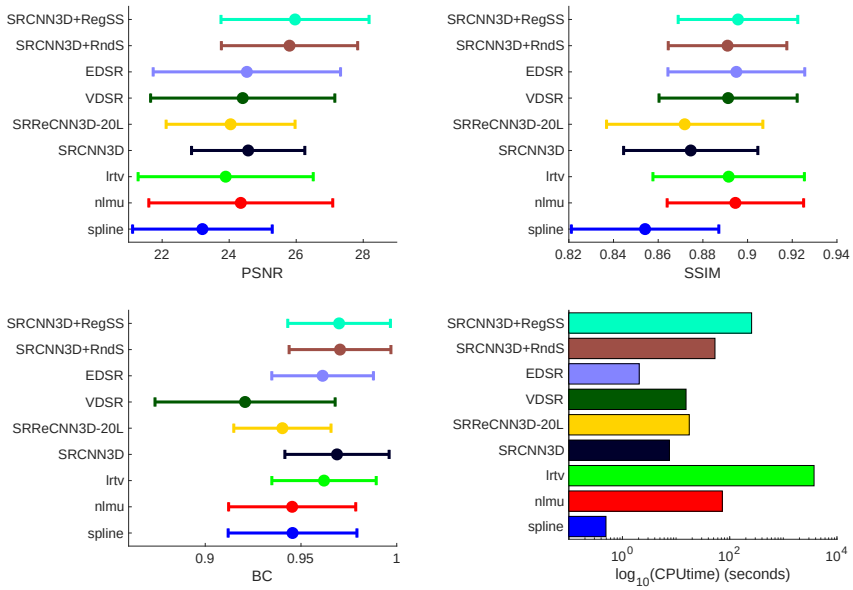
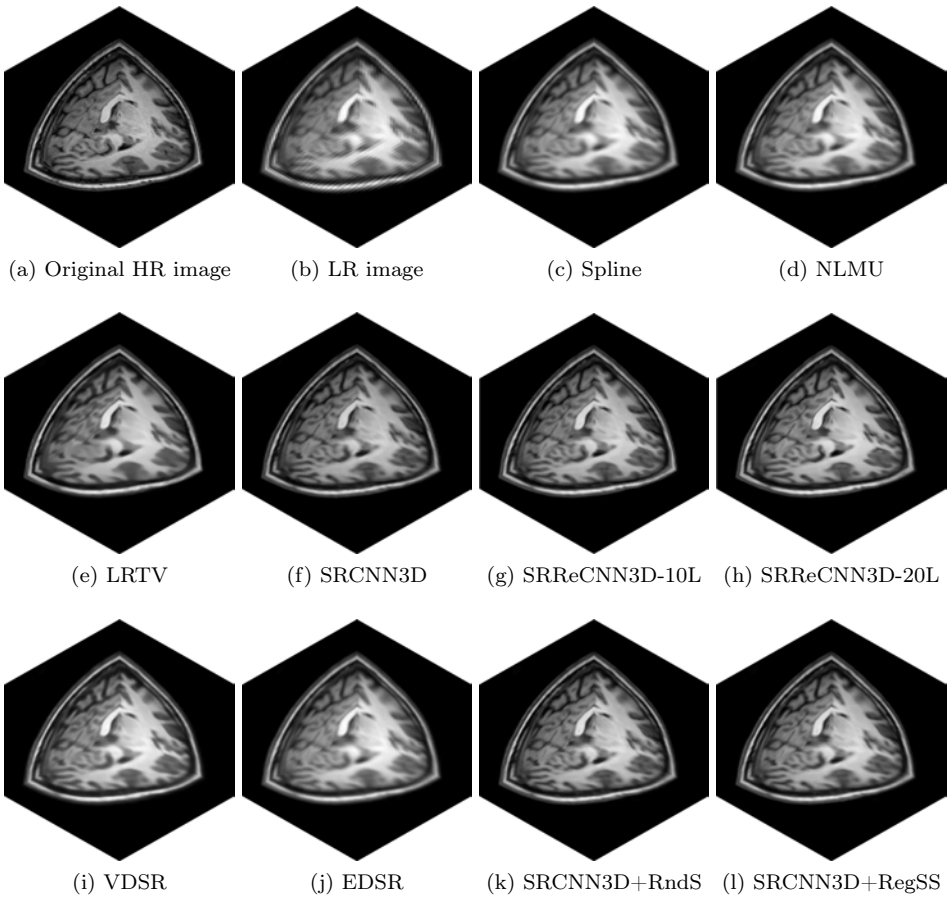


Figure 12.4: Comparison of the PSNR, SSIM, BC (higher is better) and CPU time for nine methods. Mean and standard deviation of the results for all the test images except image 5 of Kirby21 are displayed, using $\lambda = 3$.

CNN methods processes an image between 8 and 12 sec. *SRCNN3D+RegSS* takes around 3-4 minutes, which is less than *LRTV* but longer than the 53 seconds required by the random shifting method.

In Figure 12.4 are shown the outcomes using $\lambda = 3$ for the same set of images. *SRReCNN3D-10L* method is not displayed because the network was not trained for this zoom factor. In this case, *LRTV* and *NLMU* methods perform better, reaching good values of SSIM. Actually, they outperform the CNN methods *SRCNN3D* and *SRReCNN3D-20L* also for some BC values. Unlike the analysis of zoom 2, the very deep networks *VDSR* and *EDSR* performs better, improving sometimes our proposal for the SSIM measure, but with large variances in the outcomes for PSNR and BC. Analysing carefully the results, these discrepancies are originated due to their bad performance with certain images (of Kirby 21 specially). The rest of outcomes are quite suited. The models based on a shifting procedure still remain as the best ones. The highest values are obtained by the proposed *SRCNN3D+RegSS*. Finally, the ranking of the processing times is very similar to the previous ones.

In order to assess the methods from a qualitative point of view, in Figure 12.5 is depicted a three-dimensional representation of the Kirby 21 (image 11) restoration. Middle slices of each plane are shown. The smoothest results are generated by both *Spline* and *NLMU*, which do not correspond with



(a) Original HR image (b) LR image (c) Spline (d) NLMU

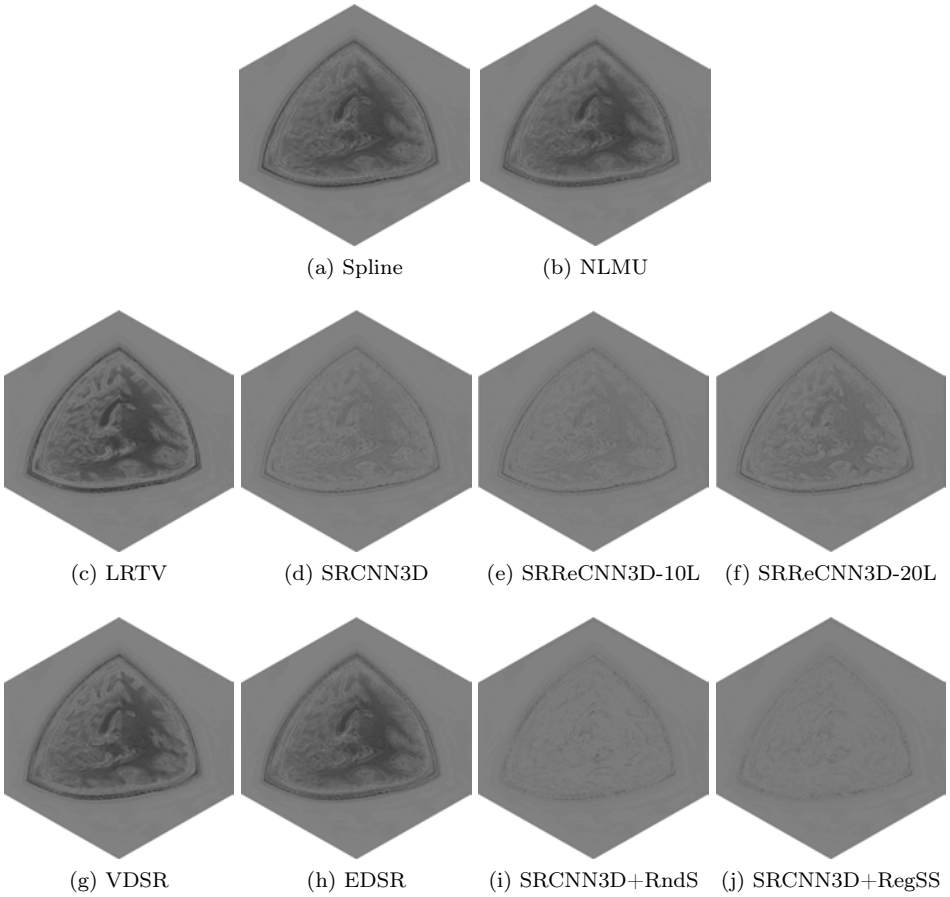
(e) LRTV (f) SRCNN3D (g) SRReCNN3D-10L (h) SRReCNN3D-20L

(i) VDSR (j) EDSR (k) SRCNN3D+RndS (l) SRCNN3D+RegSS

Figure 12.5: Qualitative results for KKI2009-11-MPRAGE T1-weighted image for each method, applied with zoom factor 2. Three-dimensional images are shown, where the XY plane corresponds to a slice of the axial view, XZ to a slice of the sagittal view and YZ to a slice of the coronal view.

the desired ideal HR image. This may indicate that they depend heavily on the degree of smoothness of the input LR image. *LRTV* is also quite different. The CNN methods yield the best restorations, with the exceptions of *EDSR*, which looks over smoothed. In order to evaluate the performance of each method and make a better differentiation, residual images were also computed and the results are shown in Figure 12.6. Here we can distinguish the best outcomes watching those images that display fewer brain structures. Raw deep learning methods yield good results, but there are still visible dark parts. However, the residual images of the shifting methods are quite gray, and actually in *SRCNN3D+RegSS* the skull is almost indistinguishable.

An example of the result obtained using zoom factor 3 from both the quantitative and qualitative points of view are shown in Figure 12.7, Fig-



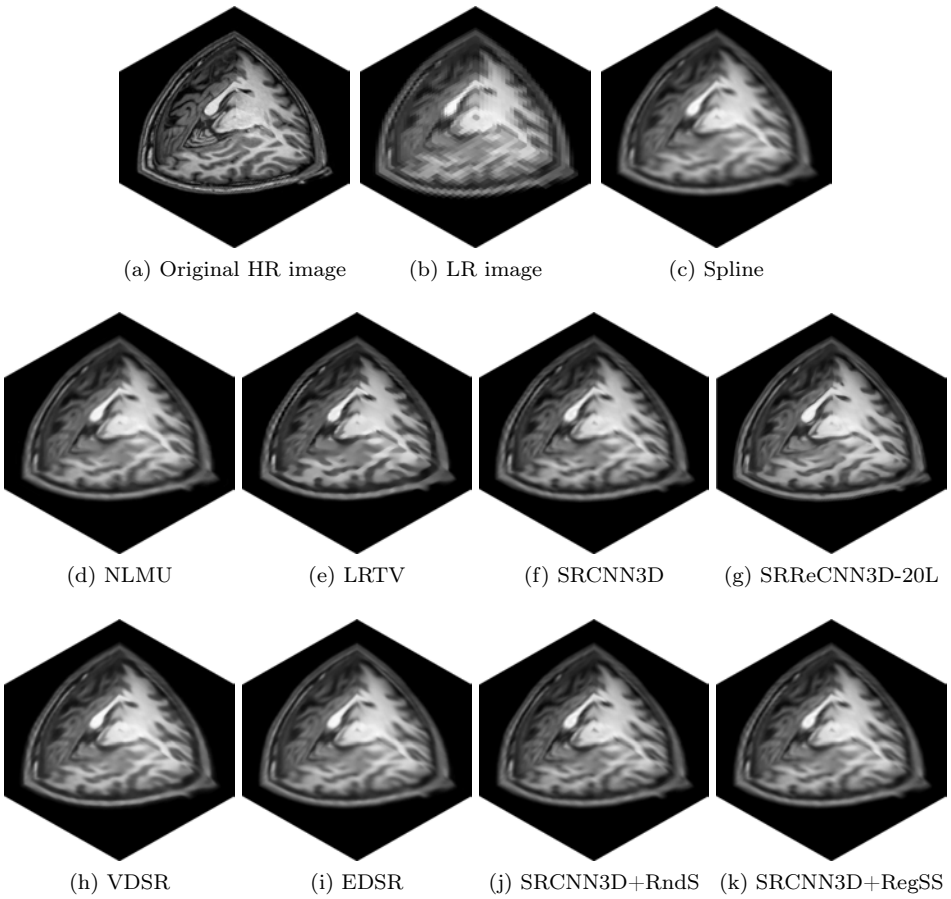
(a) Spline (b) NLMU

(c) LRTV (d) SRCNN3D (e) SRReCNN3D-10L (f) SRReCNN3D-20L

(g) VDSR (h) EDSR (i) SRCNN3D+RndS (j) SRCNN3D+RegSS

Figure 12.6: Residual images for KKI2009-11-MPRAGE T1-weighted image for each method, applied with zoom factor 2. Three-dimensional images are shown, where the XY plane corresponds to a slice of the axial view, XZ to a slice of the sagittal view and YZ to a slice of the coronal view.

ure 12.8 and Table 12.2, using the image 5 of the Kirby21 dataset. As before, *Spline* and *NLMU* are clearly worse seeing the restored image. Nevertheless, we have to focus on the residuals to be able to distinguish among the others. Here is obvious that *LRTV* still does not behave well. Comparing CNN methods, there are not great differences, but *SRCNN3D+RndS* and *SRCNN3D+RegSS* seem to yield a better outcome where the brain distortion is not severe. Table 12.2 allows us to corroborate our assumptions. Deep learning techniques perform better that traditional algorithms. The adaptation of *VDSR* and *EDSR* from 2D to 3D is not enough to outperform the other CNN methods. The inclusion of the shifting model increases the quality of the restored HR image. In some cases the differences are small, but our method yields better results in PSNR, SSIM and BC, which demonstrate



(a) Original HR image (b) LR image (c) Spline

(d) NLMU (e) LRTV (f) SRCNN3D (g) SRReCNN3D-20L

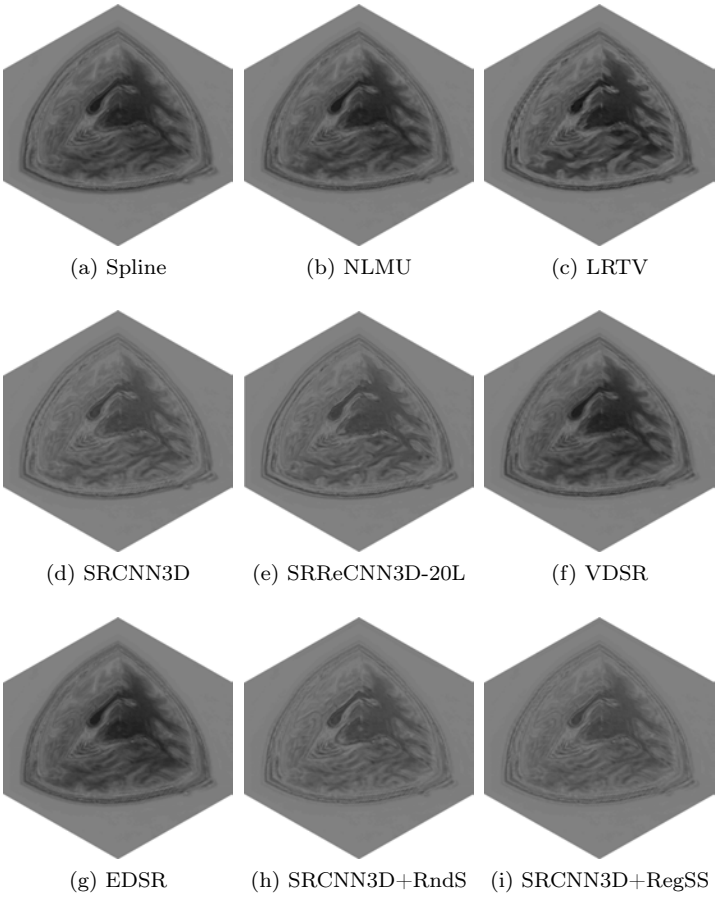
(h) VDSR (i) EDSR (j) SRCNN3D+RndS (k) SRCNN3D+RegSS

Figure 12.7: Qualitative results for KKI2009-05-MPRAGE T1-weighted image for each method, applied with zoom factor 3. Three-dimensional images are shown, where the XY plane corresponds to a slice of the axial view, XZ to a slice of the sagittal view and YZ to a slice of the coronal view.

that our proposal is effective, with an acceptable processing time.

The real T1-weighted image from CIMES was processed in Figure 12.9 and Figure 12.9. Here we can observe the notable performance of our proposal. If we focus on the residual images, we can extract two different conclusions. *SRCNN3D+RndS* and *SRCNN3D+RegSS* yield the most uniform gray residual, which means that voxel intensities of the restored image are very similar with respect to the original HR image. In addition to this, if we compare internal parts of the brain and even the neck (which is clinically irrelevant but useful for comparison purposes), the gray intensities remain stable. CNN and traditional methods have greater non uniform values of gray that could distort the original brain structures and voxel intensities.

A sagittal slice of the super-resolved OAS1_0001_MR1_mpr-1_anon



(a) Spline (b) NLMU (c) LRTV

(d) SRCNN3D (e) SRReCNN3D-20L (f) VDSR

(g) EDSR (h) SRCNN3D+RndS (i) SRCNN3D+RegSS

Figure 12.8: Residual images for KKI2009-05-MPRAGE T1-weighted image for each method, applied with zoom factor 3. Three-dimensional images are shown, where the XY plane corresponds to a slice of the axial view, XZ to a slice of the sagittal view and YZ to a slice of the coronal view.

image is shown in Figure 12.11. Here the differences are almost imperceptible except in the case of *Spline NLMU*, where there are visible smoother brain parts. Not too many differences can be appreciated among the other methods apart from the gray scale intensities shown in the residual images of Figure 12.12. in some regions of the brain. We can assure that CNN methods, including *VDSR* and *EDSR*, outperform the traditional methods. *SRCNN3D+RegSS* achieves to recover and does not remove small black spots that are distorted by the others with respect to the HR reference image.

Table 12.2: Results obtained for each method for the tested images with zoom factor 3 (higher is better for PSNR, SSIM and BC).

KKI2009-05-MPRAGE	PSNR	SSIM	BC	CPU time (sec.)
Spline	20.1656	0.8118	0.9174	**0.7436**
NLMU	20.3356	0.8396	0.9167	78.8499
LRTV	20.1216	0.8324	0.9335	4308.1375
SRCNN3D	23.6425	0.8542	0.9599	7.7902
SRReCNN3D-20L	23.4603	0.8509	0.9432	12.9448
VDSR	20.7754	0.8412	0.9080	16.6636
EDSR	20.6961	0.8459	0.9305	2.0827
SRCNN3D+RndS	24.1791	0.8703	0.9624	53.1471
SRCNN3D+RegSS	**24.3661**	**0.8771**	**0.9630**	216.7438

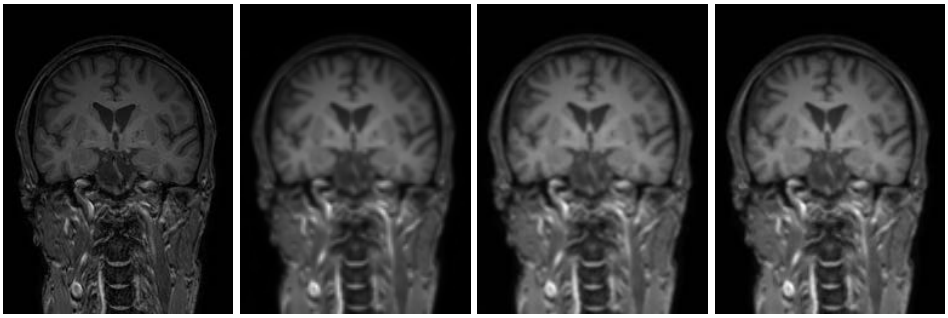
12.5 Discussion

It is important to recall that a representative metric for this kind of images is usually SSIM, because it focuses on the morphology of the brain, and not so much on the intensity values. In this sense, our proposal obtains the best SSIM values. Moreover, the high value (very close to 1) and small standard deviation we have obtained in BC are also remarkable, so we can ensure that the restored image is close to the original HR image in terms of the distribution of intensity values.

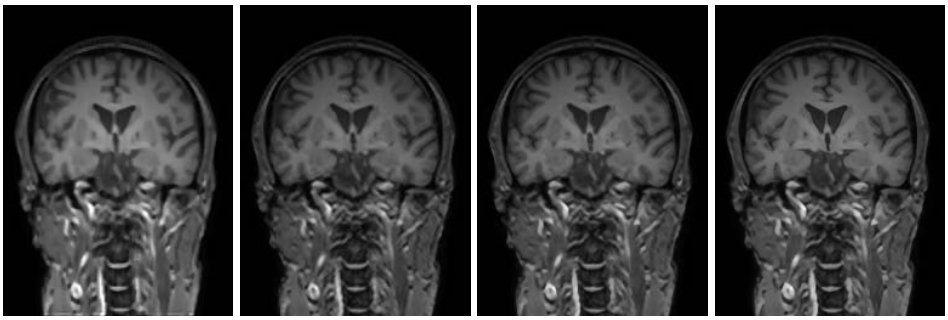
About the required CPU time, our method is not the fastest one. Nevertheless, it should be considered that a single GPU was employed when using it. The use of more GPUs simultaneously may decrease almost linearly the total time. If there are N GPUs and we compute M shifts, with $M \leq N$, it is expected a very similar performance compared to *SRCNN3D*, since the required time for the shift operations is almost insignificant.

According to the Figure 12.3 and Table 12.2, it is remarkable the performance difference between the proposed *SRCNN3D+RegSS* and the *SRCNN3D-20L*, since the latter model has a larger number of layers on its architecture. As it is commented in Section 4.3 of (Dong et al., 2016), not always "deeper is better", since it depends on the amount of data available for training and the number of epochs. In addition to this, the reported results show that the proposed shifting technique *SRCNN3D+RegSS* outperforms the compared state of the art methods and the previous proposal *SRCNN3D+RndS*. This statement is also based on the qualitative results obtained from the residual images, which can be observed in Figures 12.6, 12.8, 12.10 and 12.12. The more homogeneous and gray the image, the better the result and consequently the quality of the method. It is possible to observe that our method improves the remaining ones over the datasets studied.

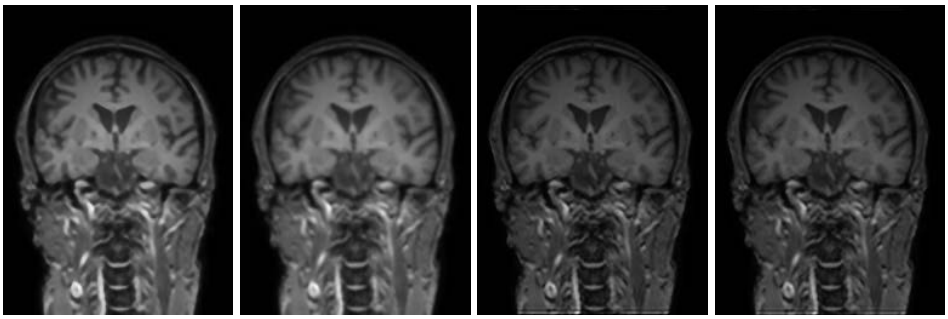
As described in Subsection 12.2.2, the advantage over the results obtained



(a) Original HR image (b) LR image (c) Spline (d) NLMU



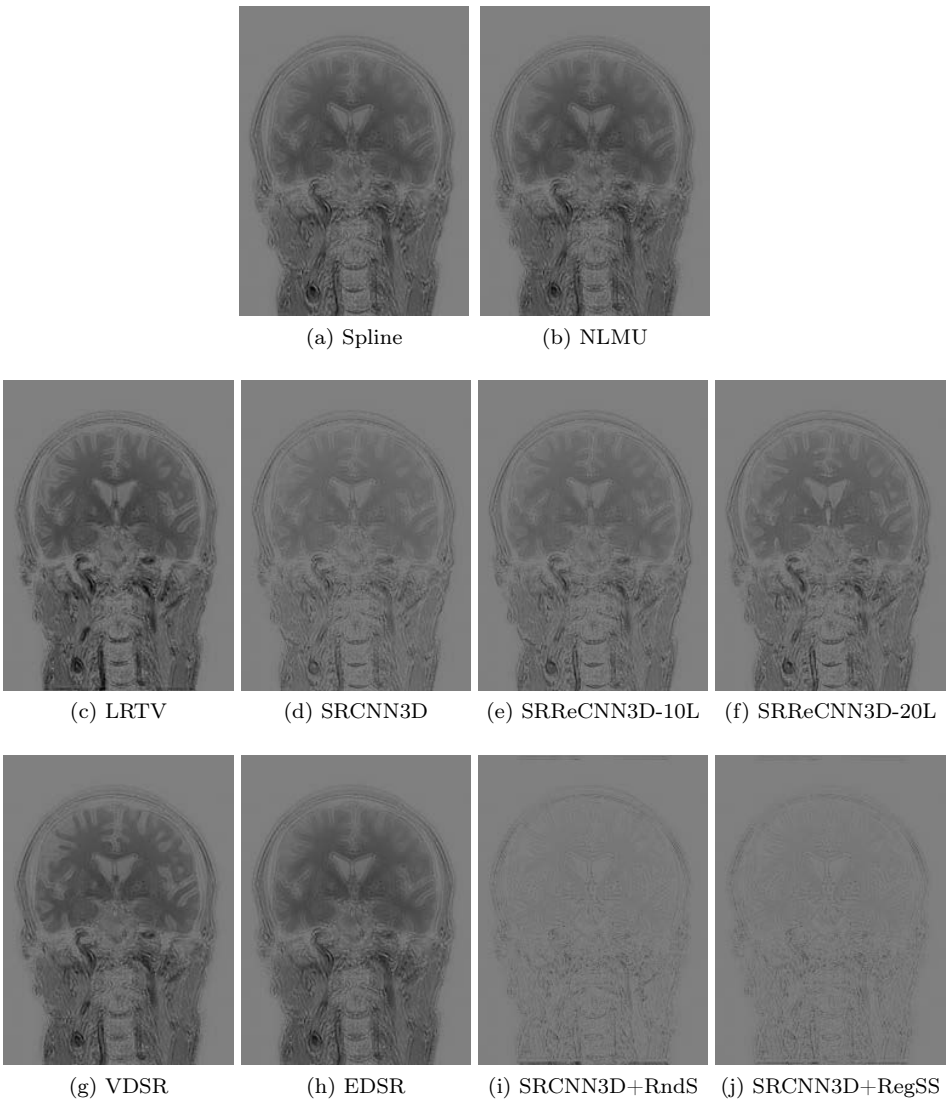
(e) LRTV (f) SRCNN3D (g) SRReCNN3D-10L (h) SRReCNN3D-20L



(i) VDSR (j) EDSR (k) SRCNN3D+RndS (l) SRCNN3D+RegSS

Figure 12.9: Qualitative results for the T1-weighted image from CIMES for each method, applied with zoom factor 2. Coronal view is shown. Second and third row display the image reconstructed by each algorithm.

by a single application of the *SRCNN3D* comes from an image filtering which is carried out on the three dimensional space of possible shifts **a**. The properties of this transformed space are worth being investigated in future works, since it is possible that careful tuning of the filtering in the transformed space could lead to even better results.



(a) Spline (b) NLMU

(c) LRTV (d) SRCNN3D (e) SRReCNN3D-10L (f) SRReCNN3D-20L

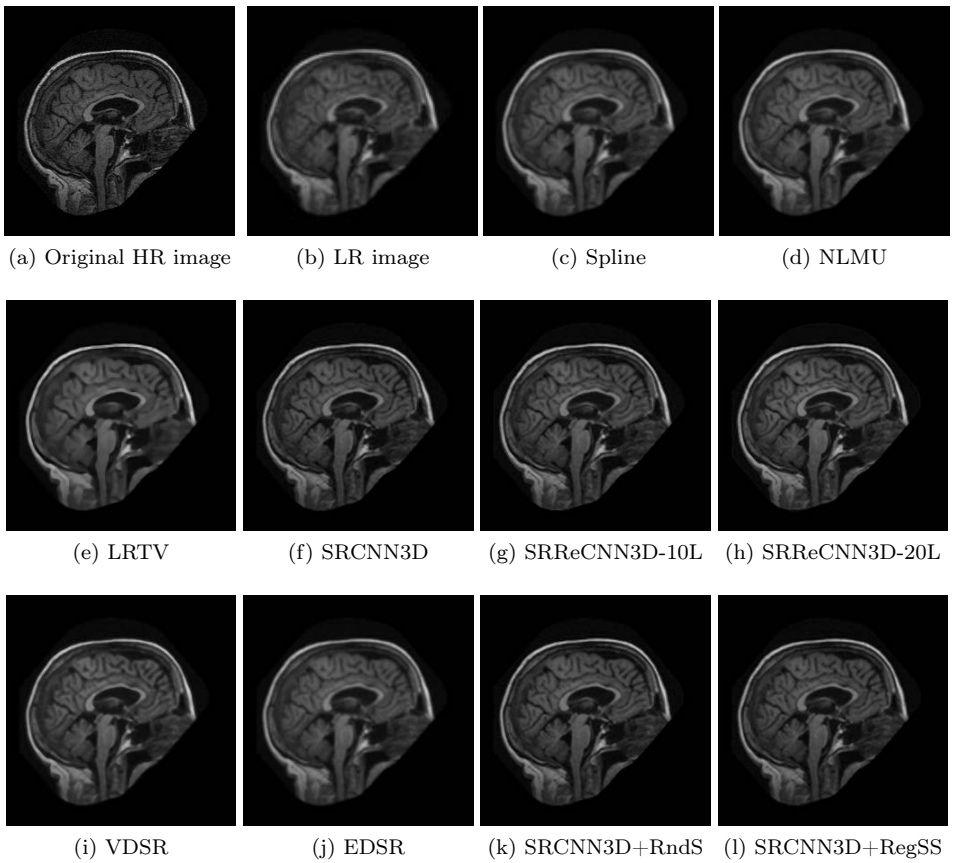
(g) VDSR (h) EDSR (i) SRCNN3D+RndS (j) SRCNN3D+RegSS

Figure 12.10: Qualitative results for the T1-weighted image from CIMES for each method, applied with zoom factor 2. Coronal view is shown. Residual images between the reconstructed and the original HR image are displayed.

12.5.1 Performance of adapted 2D models

In the present work we include comparisons with two recognised super-resolution methods for natural images, such as *VDSR* and *EDSR*. These very deep networks outperform most of the state-of-art SR algorithms, although they were designed for bidimensional natural images. These fact translates on two important limitations: the range of values and the dimensionality.

Both networks work on a fixed, 8-bit quantized range of values, that is,



(a) Original HR image (b) LR image (c) Spline (d) NLMU

(e) LRTV (f) SRCNN3D (g) SRReCNN3D-10L (h) SRReCNN3D-20L

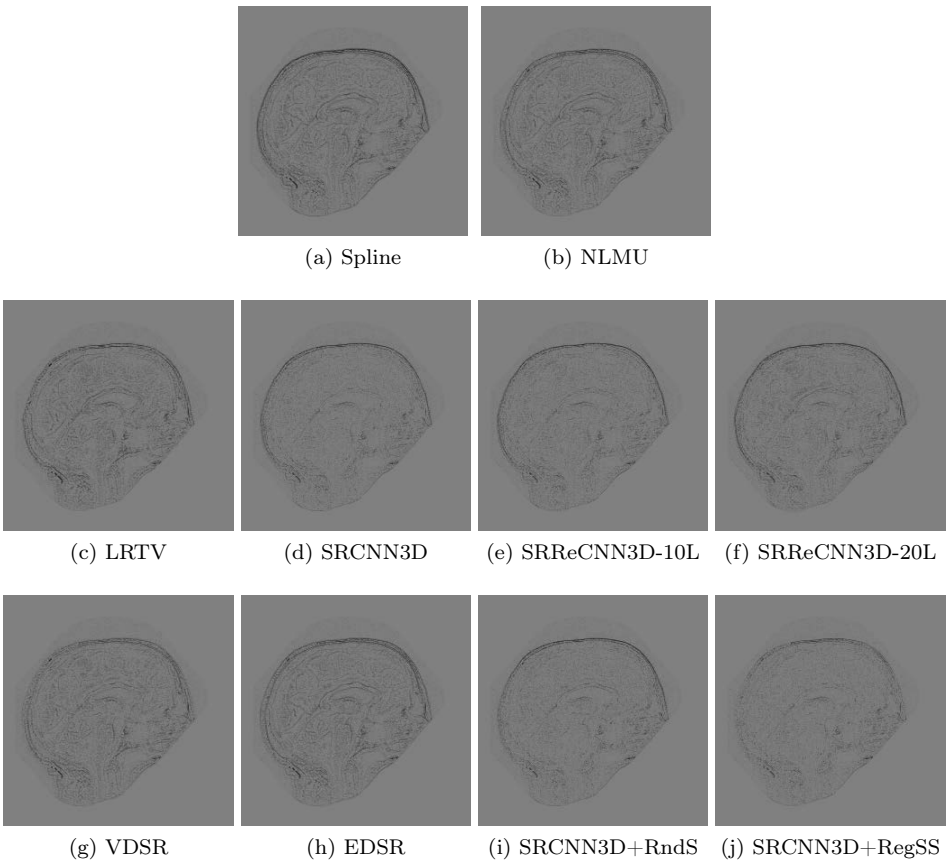
(i) VDSR (j) EDSR (k) SRCNN3D+RndS (l) SRCNN3D+RegSS

Figure 12.11: Qualitative results for OAS1_0001_MR1_mpr1_anon image for each method, applied with zoom factor 2. Reconstructed sagittal slices by each algorithm are shown.

$[0, 255]$. In particular, in the case of *VDSR*, it performs the training and the restoration of the images on the luminance channel of the $YCbCr$ color space, which is nothing more than a linear transformation of the RGB color space. In the case of magnetic resonance images, the intensity values are not quantified nor bounded, so to apply these kind of methods on these images, a normalisation is needed to transform the unbounded intensity values into the 256 possible values of the RGB space. This transformation may produce a significant loss of information that the networks cannot recover. Figures 12.5-12.10 shows the excessive smoothness of the SR images obtained by these two methods, which could have been produced by this fact.

On the other hand, we need to deal with the application of this networks on 3D images. Depending on the network, different procedures are applied. In the case of *VDSR*, we followed the steps described in both MATLAB[5]

[5] https://www.mathworks.com/help/images/single-image-super-resolution-



(a) Spline (b) NLMU

(c) LRTV (d) SRCNN3D (e) SRReCNN3D-10L (f) SRReCNN3D-20L

(g) VDSR (h) EDSR (i) SRCNN3D+RndS (j) SRCNN3D+RegSS

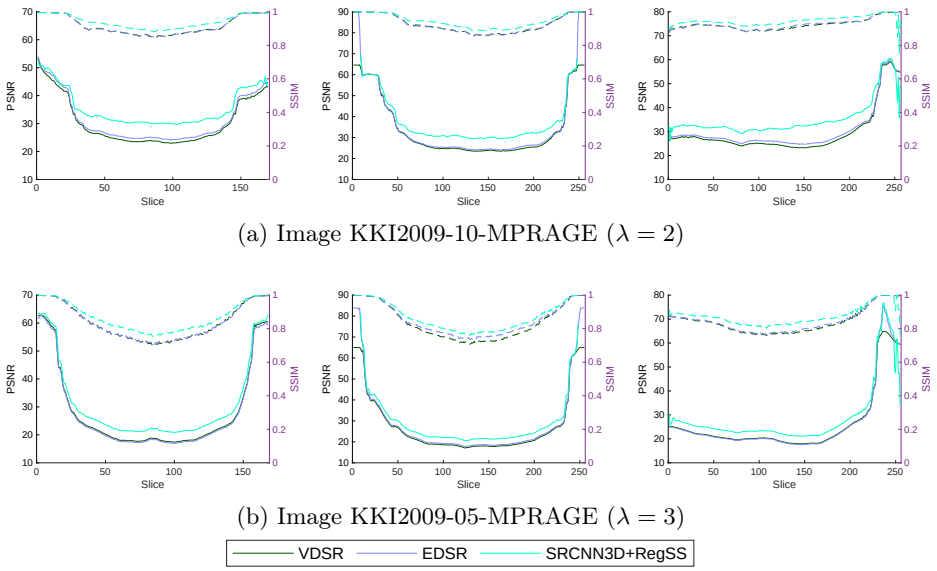
Figure 12.12: Qualitative results for OAS1_0001_MR1_mpr1_anon image for each method, applied with zoom factor 2. Residual images between the reconstructed and the original HR image are shown (sagittal slice).

and source code[6] examples. We carried out an initial upsampling to reach the desired image size and after that we apply the network restoration on each slice. *EDSR* has a different methodology. The input of this network is the LR image and the output is the HR image, so after the application on each slice, the sliced dimension must be upsampled again. That is, if the LR image has size $l \times m \times n$ and the HR has size $l' \times m' \times n'$, if we apply *EDSR* on each $m \times n$ slice, then we obtain an anisotropic $l \times m' \times n'$ image. Then we apply bicubic interpolation to restore the first dimension, i.e. to upsample from $l \times m' \times n'$ to $l' \times m' \times n'$.

In order to assess the correct application of these methods and study the effect of the dimensionality, in Figure 12.13 the PSNR and SSIM measures obtained on each slice along the three dimensions during the restoration

using-deep-learning.html

[6] https://cv.snu.ac.kr/research/VDSR/



(a) Image KKI2009-10-MPRAGE ($\lambda = 2$)

(b) Image KKI2009-05-MPRAGE ($\lambda = 3$)

Figure 12.13: Analysis of the performance of *VDSR* and *EDSR* with respect to *SRCNN3D+RegSS* for both scale factors 2 and 3. PSNR (left y-axis) is shown with a solid line and SSIM (right y-axis) with a dashed line (higher is better). Quality measures for all slices are depicted. From right to left: traversing saggital, coronal and axial axes.

procedure are shown. We depicted the performances for two different images and zoom factors, comparing our proposal with *VDSR* and *EDSR* methods. The performance of the three methods is very similar or even better for *EDSR* on boundary slices, that are those one dominated by zero voxel values. However, as we enter into the central part of the image, both PSNR and SSIM values of the 2D methods are worse than our 3D approach. In Figure 12.13b the differences are a bit smaller, and they are in concordance with Table 12.2. *VDSR* achieves a better PSNR in two of the three axes, while *EDSR* does the same in SSIM. We summarise the quality of the 3D images obtained along each axis in Table 12.3. Also, in the central columns the final PSNR and SSIM of the final reconstructed image using the straightforward strategy explained in Section 12.3 are shown.

Taking into account these analyses, the bad performance of the recognised methods *VDSR* and *EDSR* is probably due to the fact that medical images carry specific 3D structural information, and the reconstruction by a 2D model is not a natural way to keep knowledge from the image. As an example, a brain sulcus may be positioned near to the adjacent slice and also go slightly through it. In this case, this small oscillation might be difficult to distinguish from noise when a 2D model is applied, potentially resulting in noise enhancement in the SR image.

Table 12.3: Comparison of the image reconstruction of *VDSR* and *EDSR* with respect to *SRCNN3D+RegSS* analyzing the PSNR and SSIM (higher is better) of 3D SR image reconstructed on each axis.

Image	**KKI2009-10-MPRAGE** ($\lambda = 2$)			**KKI2009-05-MPRAGE** ($\lambda = 3$)		
Method	X-axis	Y-axis	Z-axis	X-axis	Y-axis	Z-axis
VDSR	26.4033/0.9313	26.7989/0.9345	26.1664/0.9318	20.9162/0.8396	20.8662/0.8395	21.2268/0.8433
	27.4724/0.9390			20.7754/0.8412		
EDSR	27.5119/0.9311	27.4537/0.9340	27.3726/0.9340	20.4648/0.8386	21.4813/0.8543	20.9599/0.8477
	27.3822/0.9335			20.6961/0.8459		
SRCNN3D+RegSS	32.7381/0.9527			24.3661/0.8771		

12.6 Conclusions

In this work, a method for magnetic resonance image super-resolution is presented. It is based on the combination of two different methodologies. Low-resolution images are processed through a convolutional neural network to perform an image restoration in order to obtain a high-resolution image. The quality of the restored images is increased by applying a regular shifting model to the input images and then recomposing them into a consensus. A variety of images of different datasets were used to evaluate the efficiency of the algorithm, obtaining successful results.

According to the used quality measures (PSNR, SSIM and BC), the proposed method has achieved better results comparing to the state of the art methods. Moreover, *SRCNN3D+RegSS* has performed the most similar voxel intensities to the original HR images, achieving the most uniform gray residuals, specially in images from CIMES.

Quantitative results show that *SRCNN3D+RegSS* overcomes other state-of-art methods, enhancing really high values of BC and measures as PSNR or SSIM, which indicates that the brain structures are not distorted. Experimental results show our proposal restores the MR images qualitatively well for different zoom factors, avoiding over-smoothing.

Future lines of research include the development of further tuned filtering methods carried out on the three dimensional shift space which has been proposed in this work.

Chapter 13

Super-resolution of 3D MRIs corrupted by heavy noise with the Median Filter Transform

It is strange that only extraordinary men make the discoveries, which later appear so easy and simple.

Georg C. Lichtenberg

Abstract: The acquisition of 3D MRIs is adversely affected by many degrading factors including low spatial resolution and noise. Image enhancement techniques are commonplace, but there are few proposals that address the increase of the spatial resolution and the noise removal at the same time. In this work such an algorithm is proposed. It tiles the 3D image space into parallelepipeds, so that a median filter is applied in each parallelepiped. The results obtained from several such tilings are then combined by a subsequent median computation. The convergence properties of the proposal are formally proved. Experimental results with both synthetic and real images are reported, where our approach outperforms its competitors for high noise levels. Moreover, it is demonstrated that our algorithm does not generate any hallucinations.

13.1 Introduction

In Magnetic Resonance Imaging (MRI), and in other medical imaging techniques such as Computerized Tomography (CT), Positron Emission Tomography (PET) or Single Photon Emission Computed Tomography (SPECT),

pixel data is affected by a number of degrading factors, notably the finite spatial resolution and noise. Lack of resolution may imply the loss of details that could be critical for medical diagnosis. Noise heavily affects image quality in some imaging modalities, and, in general, it has a negative effect on the contrast and visibility of details that could contain vital information. Given that the acquisition times and protocols are limited for practical reasons, various post-processing algorithms have been proposed to enhance resolution, and they are continuously being improved.

Self-similarity methods, as well as those based on a database, usually assume there is little noise in the images, or respond well only to low levels of noise (Yang et al., 2010). Even though (Trinh et al., 2014) has started to address this issue, it still works on a 2D-basis and requires either a low-noise single-image input or a low-noise database. A conventional approach to overcome noise-sensitivity is to apply a denoising algorithm before SR. This is not optimal, since denoising removes details and fine textures, and these effects can then be magnified in the SR step.

An interesting iterative combination of the two procedures (denoising and SR) is presented in (Singh et al., 2014), although it also works in 2D only. In order to avoid the difficulties associated to the use of an external database, it has also been proposed to use, instead, an additional HR image of the same object (Manjón et al., 2010a; Rousseau, 2008). Most Diffusion Weighted Imaging (DWI) and functional MRI brain scans are acquired together with a high resolution T1 study, which makes this method attractive for these noisy MR applications. However, it still requires initial denoising and coregistration steps before its application.

The method proposed in this work allows enhancing resolution and removing noise at the same time, working in 3D and without using any additional images. The method is a modified version of the Median Filter Transform (MFT) (López-Rubio, 2016). It is based on the computation of the median filter over randomly chosen, parallelepiped-shaped tilings of the 3D space, followed by the computation of the median of the results of these median filters.

The algorithm provides a simple implementation, and it does not make any assumptions about the noise type or the noise level, which facilitates its use for different imaging techniques. No assumptions are made about the parameters of the imaging system, such as the point spread function, and, therefore, the procedure is not affected by errors in the estimations of those parameters (Farsiu et al., 2004a). Moreover, arbitrary fractional zoom scales are allowed without the help of an additional interpolation algorithm. As our proposal does not need additional images, the applicability of the algorithm is ensured in any context. This also removes the need for longer acquisition times and the additional difficulties associated with registration problems, the need to have an external dataset, or the dependency on its features.

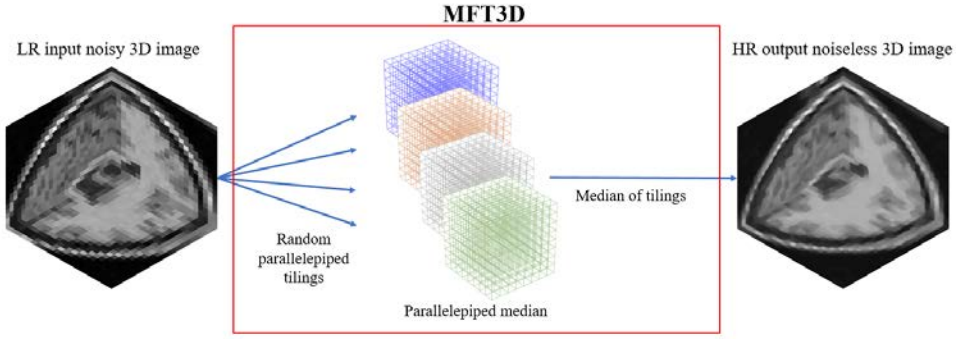


Figure 13.1: Scheme of the operation of MFT3D.

The remainder of this chapter is as follows. Section 13.2 describes the proposed transform and proves some important properties. Section 13.3 deals with the experiments on synthetic and real images. Finally, Section 13.4 is devoted to conclusions.

13.2 Material and methods

In this section the proposed approach is presented, which is called 3D Median Filter Transform (MFT3D). There are many approaches to remove the noise of 3D MRI images, but there are not so many available algorithms for 3D super-resolution applied to medical images, only for 2D images (slices), and all of them are only designed for up-sampling purposes. Here we aim at both goals using median filters. The strategy is to apply a modified version of the MFT (López-Rubio, 2016) for tridimensional images to improve the quality of the medical images and remove the noise, irrespective of the noise distribution.

First, we compute the median of over non overlapping parallelepipeds (which we will call *bins*). The output HR voxels are not necessarily at the center of the bin which is used to compute its value. Then, this procedure is repeated several times with different random template parallelepipeds to tile the space. Thus, we obtain a collection of different HR images and finally we compute the median of each HR voxel over all the images. In this way, the robustness properties of the median are preserved and the resulting HR image features a higher resolution. Our approach simplifies the original parametrization of the MFT by employing only a fixed side length instead of three independent parameters to define the parallelepiped size. The reason is that no significant differences have been found in the experimental results when we use one or more parameters to control the size of the bins. The MFT3D is defined in Subsection 13.2.1, while is convergence is formally

studied in Subsection 13.2.2. A graphical schematization of the method is depicted in Figure 13.1.

13.2.1 Definition

As we mentioned before, to compute the MFT3D a median operation must be implemented. This can be done by the classic median, but there exists another possibility, namely by using sample quantiles based on mid-distribution functions (Ma et al., 2011). For discrete distributions, these sample quantiles behave more favorably than the classical sample quantiles, which have no asymptotic normality properties.

The mid-distribution function is defined as $F_{mid}(x) = F(x) - 0.5p(x)$, where F is the cumulative distribution function and p is the probability mass function. In the case of a discrete variable X, if we denote $v_1 < \cdots < v_d$ the distinct values with the corresponding probabilities of occurrence $p_1, \ldots, p_d$, then the quantile function is defined as

$$Q(p) = F_{mid}^{-1}(p) = \begin{cases} v_1 & if\, p < p_1/2, \\ v_k & if\, p = \pi_k,\ k = 1, \ldots, d, \\ \lambda v_k + (1-\lambda) v_{k+1} & if\, p = \lambda \pi_k + (1-\lambda)\pi_{k+1}, \\ & 0 < \lambda < 1,\ k = 1, \ldots, d-1, \\ v_d & if\, p > \pi_d, \end{cases} \tag{13.1}$$

where $\pi_k = \sum_{i=1}^{k-1} p_i + p_k/2$, $k = 1, \ldots, d$, are called the mid-p-values. Now, if we use $\tilde{}$ to denote sample version empirical calculations and we denote the distinct values in the sample by $v_1 < \cdots < v_d$, with corresponding frequencies $r_1, \ldots, r_d$, then the sample quantile function $\tilde{Q}(p)$ is defined as being piecewise linear and connecting the values $\tilde{Q}\{\tilde{F}_{mid}(v_j)\}$, $j = 1, \ldots, d$. The calculation algorithm is specified with more details in (Ma et al., 2011). Since we need the sample median, we only compute $\tilde{Q}(0.5)$, and we will call it the *mid-sample median*.

Next, we define the MFT3D. Let us consider an LR image $f(\mathbf{x})$ with voxels at coordinates $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{Z}^3$. The pixel coordinates in the HR image are $\mathbf{y} = (y_1, y_2, y_3) \in \mathbb{Z}^3$. The pixel at $\mathbf{x}$ in the LR image is associated to coordinates $\alpha\mathbf{x} = (\alpha x_1, \alpha x_2, \alpha x_3) \in \mathbb{R}^3$, $\alpha \in \mathbb{R}$, $\alpha > 1$, in the HR image, where α is the zoom factor. We define the MFT3D of f as

$$\hat{f}(\mathbf{y}) = \Theta\left(\{\psi(\mathbf{y}, \mathbf{A}_1, \mathbf{b}_1), ..., \psi(\mathbf{y}, \mathbf{A}_H, \mathbf{b}_H)\}\right) \tag{13.2}$$

$$\forall i \in \{1, ..., H\},\ \psi(\mathbf{y}, \mathbf{A}_i, \mathbf{b}_i) = \Theta\left(\zeta(\mathbf{y}, \mathbf{A}_i, \mathbf{b}_i)\right) \tag{13.3}$$

where H is a constant number of tilings, $\mathbf{A}_i$ are 3×3 matrices drawn at random from a suitable distribution $p(\mathbf{A})$, $\mathbf{b}_i$ are 3×1 vectors drawn at

random from a suitable distribution $p(\mathbf{b})$, ζ is a set of voxel values of the LR image defined as

$$\zeta(\mathbf{y}, \mathbf{A}, \mathbf{b}) = \{f(\mathbf{x}) \mid \text{round}(\mathbf{A}\alpha\mathbf{x} + \mathbf{b}) = \text{round}(\mathbf{A}\mathbf{y} + \mathbf{b})\}, \tag{13.4}$$

and Θ stands for a median function which can be either the classical sample median or the mid-sample median. In both cases, the quantity $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$ is seen as a random variable from which H samples are drawn in (13.2). The set $\zeta(\mathbf{y}, \mathbf{A}, \mathbf{b})$ contains all the LR voxel values which belong to the parallelepiped where the HR voxel $\mathbf{y}$ belongs, according to the tiling of the space defined by $\mathbf{A}$ and $\mathbf{b}$.

The $\mathbf{A}$ matrix is obtained as the product of a rotation matrix $\mathbf{U}$ and a diagonal scaling matrix $\mathbf{\Lambda}$:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}, \tag{13.5}$$

$$\det(\mathbf{U}) = 1, \tag{13.6}$$

$$\det(\mathbf{A}) = \det(\mathbf{\Lambda}) = \lambda_1\lambda_2\lambda_3, \tag{13.7}$$

where λ_i are the diagonal elements of $\mathbf{\Lambda}$. The algorithm employed to generate these matrices is similar to that proposed in (López-Rubio, 2016), but the elements λ_i of the scaling matrix are generated in a different way. We define $BinSize$ as the length of the sides of the parallelepiped which defines the bin, measured in pixels in the LR image. Consequently, as we are using the same size for all sides. That is, we use cubes and not general parallelepipeds:

$$\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{BinSize}. \tag{13.8}$$

13.2.2 Convergence study

It is important to determine whether the MFT converges to any values when the number of random tilings $H \to \infty$. In this section this study is carried out, and it is also estimated how close the result is to the original noiseless HR image.

If $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$ were a continuous random variable, then a convergence rate $1/H$ would be ensured (Eliazar, 2020). But it is a discrete variable with many ties because the values of the LR voxels close to $\mathbf{y}$ are reused many times to obtain each $\psi(\mathbf{y}, \mathbf{A}_i, \mathbf{b}_i)$. The next two propositions shows results about the variance of the MFT3D for increasing finite values of H. Let $\hat{f}_H$ denote the MFT3D computed with H tilings.

Proposition 13.1. *For large enough increasing odd values of H the variance of $\hat{f}_H(\mathbf{y})$ is non increasing and upper bounded by the variance of $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$.*

That is, there is an even integer H_0 such that:

$$\forall H \in \{H_0+1, H_0+3, H_0+5, ...\}, \ \text{var}\left[\hat{f}_H(\mathbf{y})\right] \leq \text{var}\left[\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})\right] \quad (13.9)$$

$$\forall H \in \{H_0+1, H_0+3, H_0+5, ...\}, \ \text{var}\left[\hat{f}_H(\mathbf{y})\right] \geq \text{var}\left[\hat{f}_{H+2}(\mathbf{y})\right] \quad (13.10)$$

where var *stands for the variance of a random variable.*

Proof. See Proposition 1 in (López-Rubio, 2016). □

Proposition 13.2. *For large enough even values of H the variance of $\hat{f}_H(\mathbf{y})$ is lower or equal than the variance of $\hat{f}_{H-1}(\mathbf{y})$. That is, there is an even integer H_0 such that:*

$$\forall H \in \{H_0, H_0+2, H_0+4, ...\}, \ \text{var}\left[\hat{f}_H(\mathbf{y})\right] \leq \text{var}\left[\hat{f}_{H-1}(\mathbf{y})\right] \quad (13.11)$$

Proof. Again, see Proposition 2 in (López-Rubio, 2016). □

Let us denote by $z_1, \ldots, z_N$ the possible values of the discrete random variable $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$. Also, let us denote

$$q_i = P(\psi(\mathbf{y}, \mathbf{A}, \mathbf{b}) = z_1) \quad (13.12)$$

where $q_i > 0$, and

$$\sum_{i=1}^{N} q_i = 1. \quad (13.13)$$

Two cases must be distinguished. If the following condition holds:

$$\exists j \in \{1, ..., N-1\} \ \sum_{i=1}^{j} q_i = \sum_{i=j+1}^{N} q_i = \frac{1}{2} \quad (13.14)$$

then both z_j and z_{j+1} are medians of $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$. This is a degenerate case that is unlikely to occur in practice, since if the q_i are chosen at random from any continuous distribution, then the probability that (13.14) holds is zero. Proposition 13.3 studies this case.

If (13.14) does not hold, then there is a unique median $m = z_{j+1}$ with $j \in \{0, 1, ..., N-1\}$, where:

$$\left(\sum_{i=1}^{j} q_i < \frac{1}{2}\right) \text{ and } \left(\sum_{i=1}^{j+1} q_i > \frac{1}{2}\right) \quad (13.15)$$

This second case, which is the one to be expected in practice, is studied by Proposition 13.4.

Proposition 13.3. *If (13.14) holds then the distribution of* $\hat{f}_H(\mathbf{y})$ *converges to the two point uniform distribution on the medians* z_j *and* z_{j+1} *of* $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$ *as* $H \to \infty$.

Proof. See Proposition 3 in (López-Rubio, 2016). □

Proposition 13.4. *If (13.14) does not hold then the distribution of* $\hat{f}_H(\mathbf{y})$ *converges to the unique median of* $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$ *as* $H \to \infty$.

Proof. See Proposition 4 in (López-Rubio, 2016). □

These two propositions proves that the MFT converges to the median of the distribution of the parallelepiped medians $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$, which are the median of the values of the voxels inside the parallelepiped defined by $\mathbf{A}$ and $\mathbf{b}$. This median can be unique in the general case (Proposition 13.4) or two medians in the degenerate case (Proposition 13.3).

If we consider the simple case of a constant image corrupted by additive Gaussian noise, i.e., $f(x) \sim N(\mu, \sigma)$, the following proposition proves the asymptotic convergence of the MFT3D to the noiseless image provided that all the medians are computed over sets with odd cardinals.

Proposition 13.5. *Let* f *be a a constant image with value* μ *corrupted by additive Gaussian noise of standard deviation* σ, $f(\mathbf{x}) \sim N(\mu, \sigma)$. *If* H *is odd and the number of elements of* $\zeta(\mathbf{y}, \mathbf{A}_i, \mathbf{b}_i)$ *is also an odd number* $2J+1$ *for all* $i \in \{1, ..., H\}$, *then the MFT converges to the noiseless image value* μ *as* $H \to \infty$ *and* $J \to \infty$.

Proof. See Proposition 5 in (López-Rubio, 2016). □

Next the validity of the MFT3D for subvoxel value estimation is studied.

Proposition 13.6. *The probabilities* q_i *are continuous functions of* $\mathbf{y}$.

Proof. In order to study the continuity of q_i as a function of $\mathbf{y}$ we must show that $|q_i(\mathbf{y}+\boldsymbol{\epsilon}) - q_i(\mathbf{y})| \to 0$ as $\|\boldsymbol{\epsilon}\| \to 0$. Let us consider that $\mathbf{y}$ and $\mathbf{y}+\boldsymbol{\epsilon}$ belong to different parallelepipeds if and only if $\text{round}(\mathbf{A}\mathbf{y}+\mathbf{b}) \neq \text{round}(\mathbf{A}(\mathbf{y}+\boldsymbol{\epsilon})+\mathbf{b})$. Since the values of $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$ and $\psi(\mathbf{y}+\boldsymbol{\epsilon}, \mathbf{A}, \mathbf{b})$ can only be different if the parallelepipeds are different, it follows that:

$$|q_i(\mathbf{y}+\boldsymbol{\epsilon}) - q_i(\mathbf{y})| \leq P(\text{round}(\mathbf{A}\mathbf{y}+\mathbf{b}) \neq \text{round}(\mathbf{A}(\mathbf{y}+\boldsymbol{\epsilon})+\mathbf{b})) \tag{13.16}$$

From (13.5) and (13.8) we find that the norm of $\mathbf{A}$ is upper bounded:

$$\|\mathbf{A}\| = \|\mathbf{U}\boldsymbol{\Lambda}\| \leq \|\mathbf{U}\| \|\boldsymbol{\Lambda}\| = \|\boldsymbol{\Lambda}\| = \frac{1}{BinSize} \tag{13.17}$$

where $\|\cdot\|$ stands for the spectral norm of a matrix and $\|\mathbf{U}\| = 1$ because $\mathbf{U}$ is orthogonal, and the norm of $\mathbf{\Lambda}$ is its largest element because $\mathbf{\Lambda}$ is a diagonal matrix with non-negative entries. Consequently $\|\mathbf{A}\boldsymbol{\epsilon}\|$ is also bounded:

$$\|\mathbf{A}\boldsymbol{\epsilon}\| \leq \|\mathbf{A}\| \, \|\boldsymbol{\epsilon}\| \leq \frac{1}{BinSize} \|\boldsymbol{\epsilon}\| \tag{13.18}$$

On the other hand:

$$\|(\mathbf{A}(\mathbf{y} + \boldsymbol{\epsilon}) + \mathbf{b}) - (\mathbf{A}\mathbf{y} + \mathbf{b})\| = \|\mathbf{A}\boldsymbol{\epsilon}\| \tag{13.19}$$

Then it turns out that $\|(\mathbf{A}(\mathbf{y} + \boldsymbol{\epsilon}) + \mathbf{b}) - (\mathbf{A}\mathbf{y} + \mathbf{b})\| \to 0$ as $\|\boldsymbol{\epsilon}\| \to 0$. Since $\mathbf{b}$ is uniformly distributed on $[0,1]^3$, the fractional part of $\mathbf{A}\mathbf{y} + \mathbf{b}$ is also uniformly distributed on $[0,1]^3$ irrespective of the values of $\mathbf{A}$ and $\mathbf{y}$. Consequently,

$$\begin{aligned} P(\text{round}(\mathbf{A}\mathbf{y} + \mathbf{b}) \neq \text{round}(\mathbf{A}(\mathbf{y} + \boldsymbol{\epsilon}) + \mathbf{b})) \\ \leq 8 \|(\mathbf{A}(\mathbf{y} + \boldsymbol{\epsilon}) + \mathbf{b}) - (\mathbf{A}\mathbf{y} + \mathbf{b})\| \end{aligned} \tag{13.20}$$

so that the probability in (13.16) tends to 0 as $\|\boldsymbol{\epsilon}\| \to 0$, which implies that q_i is a continuous function of $\mathbf{y}$. □

From this result we can deduce that the functions of the form $\sum_{i=1}^{j} q_i$ are continuous, since they are sum of continuous functions. The MFT3D defines regions so that inside a region it always converges to the same value as $H \to \infty$. This can be derived from (13.15) and Proposition 13.4. It means that this method is valid for subvoxel value estimation, because as $H \to \infty$, a subvoxel and all its immediate neighbors lie in the same region and converges to the same value.

Lastly, the following proposition proves the convergence in distribution of the Median Filter Transform when the mid-sample median is employed. It must be emphasized that the value that it converges to might not be the original noiseless HR value. This is due to the down-sampling process, since the LR image loses some information, so that the noise-free HR image cannot be recovered because there are many HR images that correspond to the same noisy LR image.

Proposition 13.7. *Let $\hat{f}_H$ be the MFT3D computed for an image $\mathbf{y}$ using the mid-sample median, and consider the distribution of the discrete random variable $\psi(\mathbf{y}, \mathbf{A}, \mathbf{b})$, from which we have H independent observations. Let $z_1 < \cdots < z_N$ be the support and $q_1, \ldots, q_N$ the corresponding probabilities of the distribution. Define $\pi_k = \sum_{i=1}^{k-1} q_i + q_k/2$. Then,*

$$\sqrt{H}(\hat{f}_H - \tilde{f}) \frac{1}{2} \frac{q_{k+1} + q_{k+2}}{z_{k+2} - z_{k+1}} \to N(0, \sigma^2) \text{ in distribution as } H \to \infty, \tag{13.21}$$

$$\sigma^2 = \frac{1 - \{1 - (\lambda - 1)^2\} q_{k+1} - (1 - \lambda^2) q_{k+2}}{4}, \tag{13.22}$$

where $0 < \lambda < 1$ *and for some* $k \in \{0, \ldots, N-2\}$,

$$0.5 = \lambda\pi_{k+1} + (1-\lambda)\pi_{k+2}, \tag{13.23}$$

$$\tilde{f} = \lambda z_{k+1} + (1-\lambda) z_{k+2} \tag{13.24}$$

Proof. It follows immediately from the case 3 of Theorem 2 in (Ma et al., 2011), where we take $p = 0.5$. The other cases of Theorem 2 in (Ma et al., 2011) are extreme cases that have a negligible probability of occurrence for real medical images. □

This result demonstrates that the MFT3D converges as $H \to \infty$ if we are using the mid-sample median to compute the HR image. If the classic sample median were used, it would be difficult to see whether the method converges to anything.

13.3 Experiments and results

In this section, the results of our SR experiments are presented. First we describe the competing methods used to compare (Subsection 13.3.1), and then we explain how we carry out the experiments and the performance measures (Subsection 13.3.2). Finally, we present our results with synthetic data (Subsection 13.3.3), and with real data (Subsection 13.3.4). The source code and a demo of our method are publicly available[1].

13.3.1 Competing methods

Since there are not many developed SR algorithms for three dimensional MR images, and most of them do not have available code, we have used MR denoising algorithms as the first step, and then we resample the denoised image to the desired resolution using a tridimensional cubic spline interpolation as implemented in MATLAB (Mathworks Inc.).

The following denoising methods were used to compare the proposed method:

- NLM3D: estimation of the noiseless pixel value by weighted averaging over a large portion of the input image, where the weights are based on the relative similarities among the neighbor pixels and that one to be estimated (Coupe et al., 2008; Liu et al., 2010; Manjón et al., 2010c).
- WSM: is a fully automatic 3D blockwise version of the nonlocal means filter with wavelet subbands mixing, based on a multiresolution approach for improving the quality of image denoising filter (Coupé et al., 2008).

[1] The source code and demo will be published in case of acceptance.

- ODCT3D: based on a three-dimensional moving-window discrete cosine transform hard thresholding, it is a method that take advantage of the high compressibility (sparseness) of MRI data, allowing a very efficient noise reduction (Manjón et al., 2012).

- PRI-NLM3D: based on a three-dimensional rotationally invariant version of the well-known nonlocal means filter, in this method the similarities between voxels and patches are computed from the already denoised image using the ODCT3D method and then applied to denoise the original noisy image (Manjón et al., 2012).

- BD4M: implements the grouping and collaborative filtering paradigm, where mutually similar d-dimensional cubes of voxels are stacked together and jointly filtered in transform domain, which exploits the local correlation present among voxels in each cube and the nonlocal correlation between the corresponding voxels of different cubes (Maggioni et al., 2013).

In addition to these, we only have found one SR algorithm to be applied for MR images:

- NLMU: this method uses a data-adaptive patch-based reconstruction in combination with a subsampling coherence constraint. The presence of noise is minimized by the application of a filter using WSM method (Manjón et al., 2010b).

In fact, this method is a variant of the preceding ones. The difference lies in the up-sampling method used after the denoising step, which is done by interpolation.

13.3.2 Experiment details

Dimensions of the original images were trimmed to be congruent with the zoom factor. LR images were obtained down-sampling the HR images by two steps (Dong et al., 2013): first, the HR image was filtered using a 3-D Gaussian smoothing kernel with unit standard deviation. Then, LR image of the required size is obtained by cubic spline interpolation. For synthetic experiments, LR images are generated after noise is added to HR images.

Furthermore, all experiments with our method were carried out using $H = 150$ median filters. Integer zoom factors $\alpha \in \{2, 3, 4\}$ and fractional zoom factors $\alpha \in \{2.5, 3.5\}$ have been considered. None of the competitors except NLMU use the zoom factor because they are denoising methods, while the super-resolution step is done by the bicubic interpolation algorithm. In the case of NLMU, the available code only allows integers factors, so when we are in the case of a non-integer zoom we followed the author's instructions.

Table 13.1: Parameter selection for MFT3D: *BinSize* value for each image type and each zoom factor.

	$\alpha = 2$	$\alpha = 2.5$	$\alpha = 3$	$\alpha = 3.5$	$\alpha = 4$
T1	1.6	2.55	3.15	3.6	4.1
T2	1.6	2.2	2.75	3.2	3.75
PD	1.6	2.4	2.95	3.4	3.95

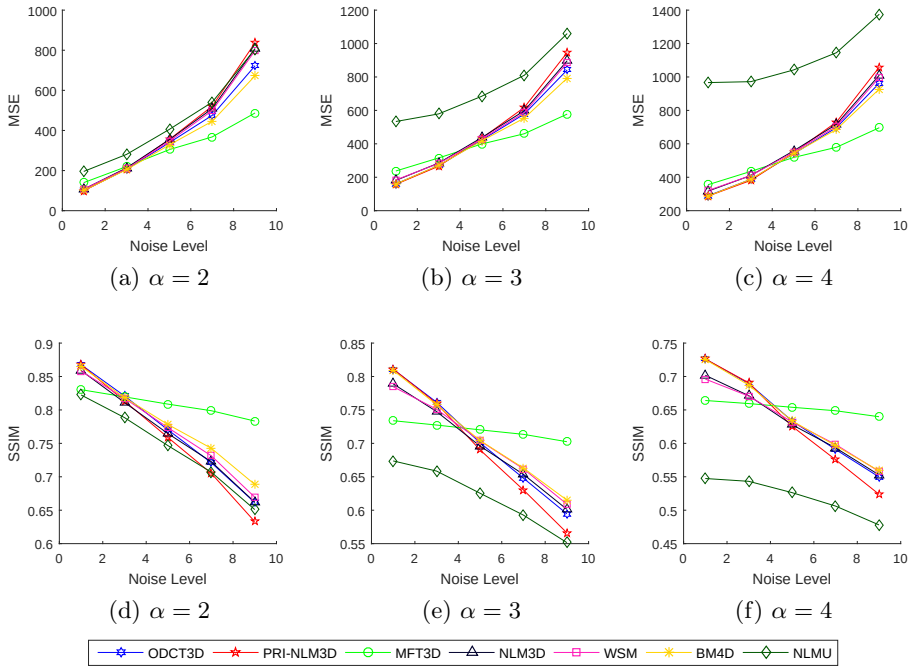
The desired image is obtained by up-sampling the images to the closest superior integer factor and then subsampling to the desired resolution.

We have carried out a parameter optimization for MFT3D prior to the main experiments, so that they have precomputed values of their parameters for each zoom factor, and each image type (T1, T2 or PD). The comparisons are fair because each competing method has precomputed values of their adjustable parameters for each zoom factor which is used for all images, noise types, and noise levels. These optimized parameters have been used as they are. This way parameter values of the proposed method which are adequate for different images and zoom factors have been selected and reported in Table 13.1.

Two quality measures were used to evaluate the proposed approach: the first was the Mean Squared Error (MSE) metric (lower is better), which is commonly used in image processing, and the second was the Structural Similarity Index Measure (higher is better), which focuses on structural similarities between images and is more consistent in human images:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{13.25}$$

where $\mu_x$and μ_y are the mean value of images x and y, σ_x and σ_y are the standard deviation of images x and y, σ_{xy} is the covariance of x and y, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ (L is the dynamic range, $k_1 = 0.01$ and $k_2 = 0.03$). As suggested by (Wang et al., 2004), the SSIM was locally estimated using a Gaussian kernel of $3 \times 3 \times 3$ voxels. Finally, the mean value of all the local estimations was used as a quality metric. To do a realistic comparison, both measures were estimated only in the head tissue, i.e., for an image I with values in the range $[0, 255]$, we use $\hat{I} = \{v \in I : \ v > 10 \cdot D/255\}$, where D is the peak of I. In addition to this, CPU times were computed to do a simple comparison between methods. The reported experiments have been carried out on a 64-bit Personal Computer with an eight-core Intel i7 3.60GHz CPU, 32 GB RAM and standard hardware. The implementation of our approach does not use any GPU resources.



(a) $\alpha = 2$ (b) $\alpha = 3$ (c) $\alpha = 4$

(d) $\alpha = 2$ (e) $\alpha = 3$ (f) $\alpha = 4$

Figure 13.2: Quantitative results for the integer zoom factor experiments using T1-weigthed image from Brainweb database. The top row shows the MSE metric, while the bottom row shows the SSIM metric.

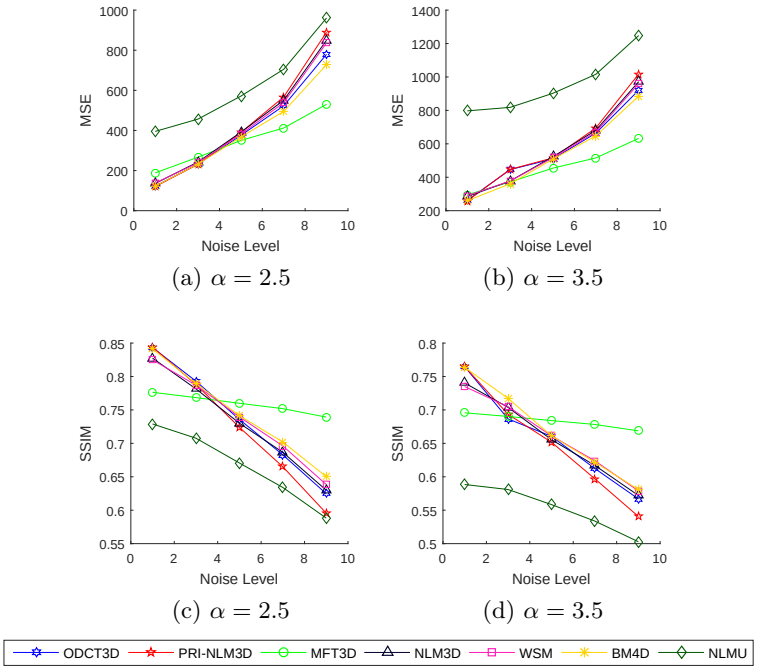
On the other hand, we need to do a qualitative evaluation or MRI denoising methods by examination of the difference between the input noisy image t and the restored image $\hat{r}$ obtained by each method:

$$\hat{e} = t - \hat{r} \tag{13.26}$$

We can compare the residual images with the ideal residual image, which correspond to the difference between the input noisy image t and the original noiseless image r:

$$e = t - r \tag{13.27}$$

For sake of clarity in the visual examination of the image, a constant was added to the residual images. In the case of real images (noisy images) we do not have available their ground truths (noiseless images), so we cannot compute the ideal residual image. However, to assess the quality of the restoration, the residual image must have little anatomical information and regions with zero values in the restored image should be darker, i.e., values near to zero, in the residual image.



(a) $\alpha = 2.5$ (b) $\alpha = 3.5$

(c) $\alpha = 2.5$ (d) $\alpha = 3.5$

Figure 13.3: Quantitative results for the fractional zoom factor experiments using T1-weigthed image from Brainweb database. The top row shows the MSE metric, while the bottom row shows the SSIM metric.

13.3.3 Synthetic data

The freely available open access BrainWeb benchmark database was used to carry on the experiments of this section (McConnell Brain Imaging Centre, 2006). T1, T2 and PD weighted volumes of $181 \times 217 \times 181$ voxels and voxel resolution of $1 \times 1 \times 1 mm^3$ were used, with zero noise and zero non-uniformity. Their original values are in the range $[0, 4095]$ (12 bit precision), which have been normalized to $[0, 1]$ in order to process the image. Then, they have been rescaled to $[0, 255]$ to measure the quality.

Five Rician noise levels have been tested: 1%, 3%, 5%, 7%, 9%. The noisy images have also been obtained from the Brainweb database. The noise percentage is relative to the average real and imaginary values of the overall brightest tissue class. Noise is generated using a pseudorandom Gaussian noise field which is added to both the real and imaginary components before the final magnitude value of the simulated image is computed.

Quantitative results for the tested integer and fractional zoom factors, and noise levels are reported in Figures 13.2-13.3. As seen in both figures, our proposed MFT3D method obtain the best results when moderate and heavy noise is present in the image ($\geq 5\%$). This happens both at the voxel

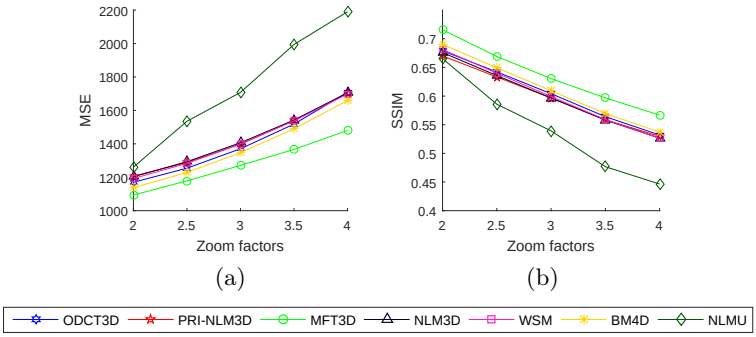


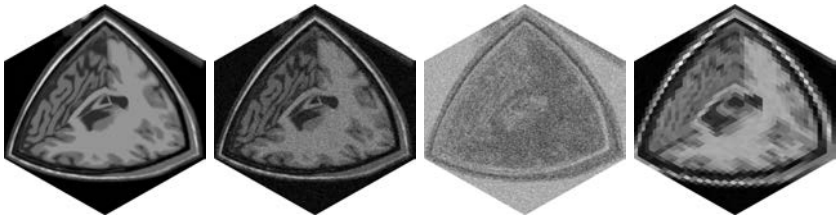
Figure 13.4: Quantitative results for Brainweb data varying the zoom factor. Average of the MSE and SSIM measures over the T1, T2 and PD images, and all noise levels.

level (MSE) and at the local structure level (SSIM). In the case of lower noise levels, our method produces slightly larger errors than the competitors. As expected, all the Non-Local Means (NLM) based filters have a similar behavior and almost the same values, they suffer from its lack of adaptation to the properties of Rician noise. The pure SR competitor, NLMU, obtains the worst results. The reason of that may be caused due to the application of a spline interpolation before applying a regularization, which can smooth the final intensity values.

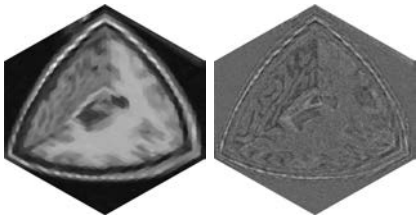
In Figure 13.4, we summarize in two graphics the results for the employed images from Brainweb dataset. For each zoom factor $\alpha \in \{2,\ 2.5,\ 3,\ 3.5,\ 4\}$, we calculated the average over all the images and noise levels we have used. As it can be seen, our method clearly overcomes the competitors for all the zoom factors analyzed, both for MSE and SSIM measures, which confirms the better performance of MFT3D.

This can also be seen with the qualitative evaluation. Figures 13.5-13.7 show tridimensional sections of the resulting images of each evaluated method, with their respective residual images, for T1, T2 and PD scans. The axial view is represented in the XY plane, the sagittal view in the YZ plane and the coronal view in the XZ plane. It can be seen that, for the three types of images, MFT3D yields more definite restored images, while the competing methods yield smoother images, where part of the anatomical information is lost. In addition, we can compute the ideal residual image since the original noiseless image is available. The comparison shows that pure SR methods, MFT3D and NLMU apparently remove a higher amount of noise, since the obtained images are darker.

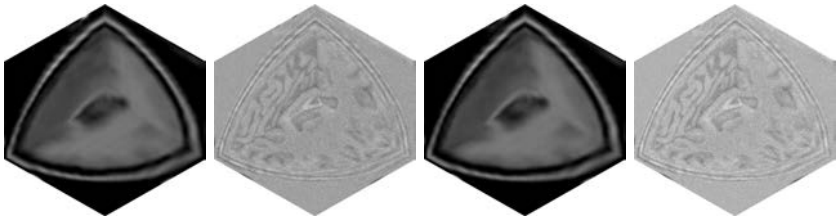
An axial slice of T1 Brainweb image is presented in Figure 13.8. It shows the original noiseless image, the generated LR noisy image, and the HR results for our method and one of the competitors (BD4M) based on



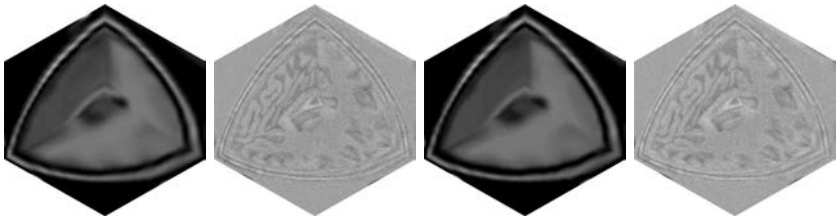
(a) HR noiseless image (b) 9% Rician noise added (c) ideal residual (d) LR input image



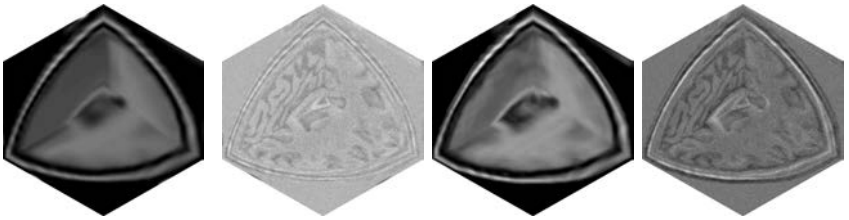
(e) MFT3D result (f) MFT3D residual



(g) NLM3D result (h) NLM3D residual (i) WSM result (j) WSM residual

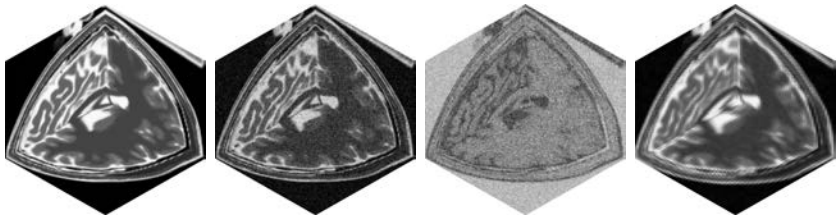


(k) ODCT3D result (l) ODCT3D residual (m) PRI_NLM3D result (n) PRI_NLM3D residual

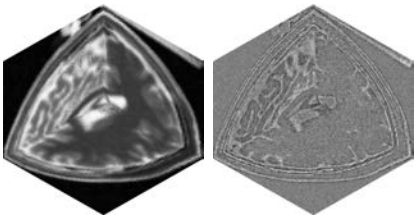


(o) BD4M result (p) BD4M residual (q) NLMU result (r) NLMU residual

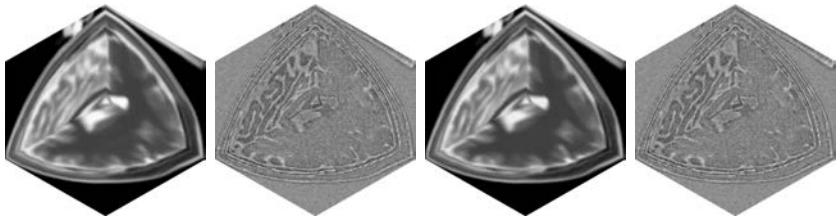
Figure 13.5: Qualitative results for BrainWeb T1 weighted synthetic image, $\alpha = 4$.



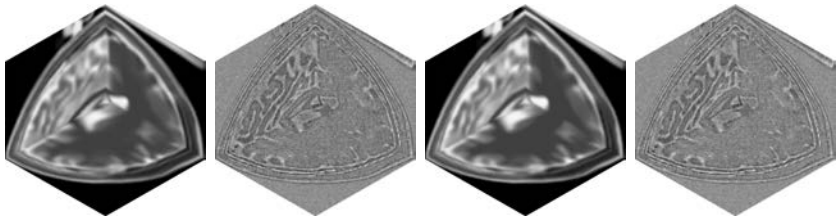
(a) HR noiseless image (b) 7% Rician noise added (c) ideal residual (d) LR input image



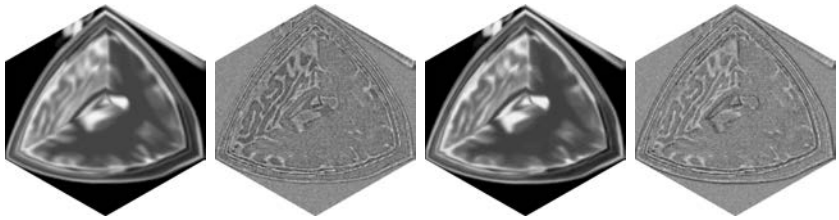
(e) MFT3D result (f) MFT3D residual



(g) NLM3D result (h) NLM3D residual (i) WSM result (j) WSM residual

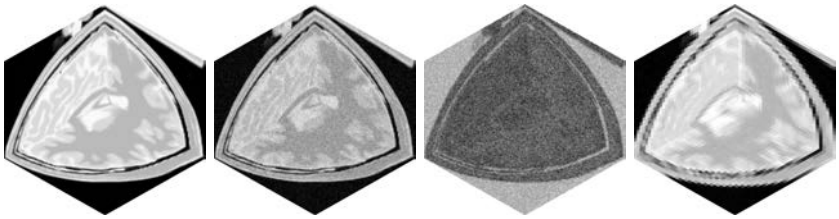


(k) ODCT3D result (l) ODCT3D residual (m) PRI_NLM3D result (n) PRI_NLM3D residual

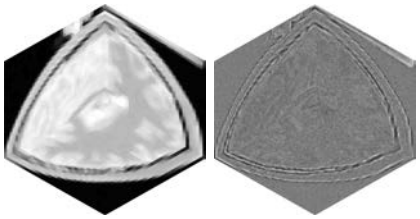


(o) BD4M result (p) BD4M residual (q) NLMU result (r) NLMU residual

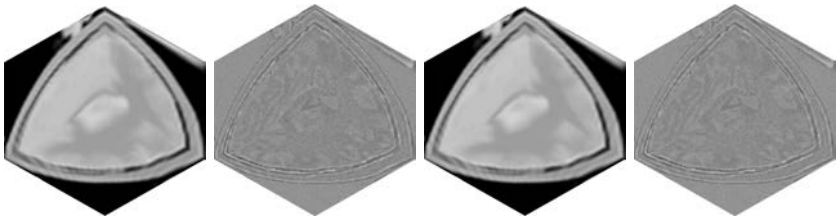
Figure 13.6: Qualitative results for BrainWeb T2 weighted synthetic image, $\alpha = 2$.



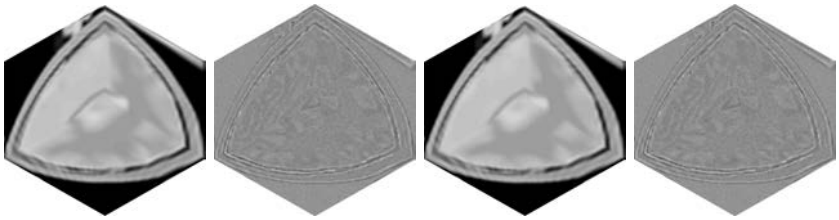
(a) HR noiseless image (b) 5% Rician noise added (c) ideal residual (d) LR input image



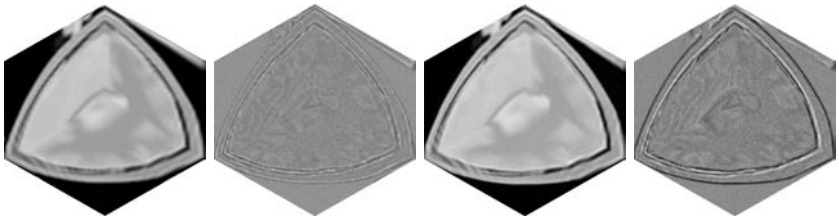
(e) MFT3D result (f) MFT3D residual



(g) NLM3D result (h) NLM3D residual (i) WSM result (j) WSM residual

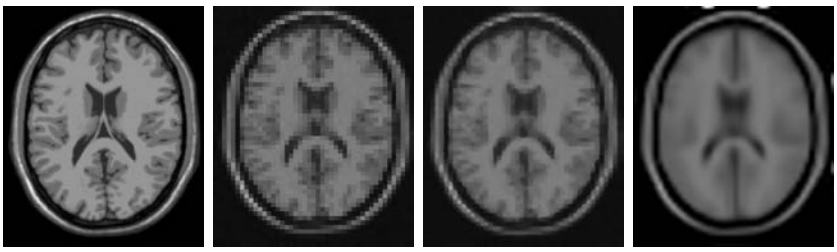


(k) ODCT3D result (l) ODCT3D residual (m) PRI_NLM3D result (n) PRI_NLM3D residual



(o) BD4M result (p) BD4M residual (q) NLMU result (r) NLMU residual

Figure 13.7: Qualitative results for BrainWeb PD weighted synthetic image, $\alpha = 3$.



(a) (b) (c) (d)

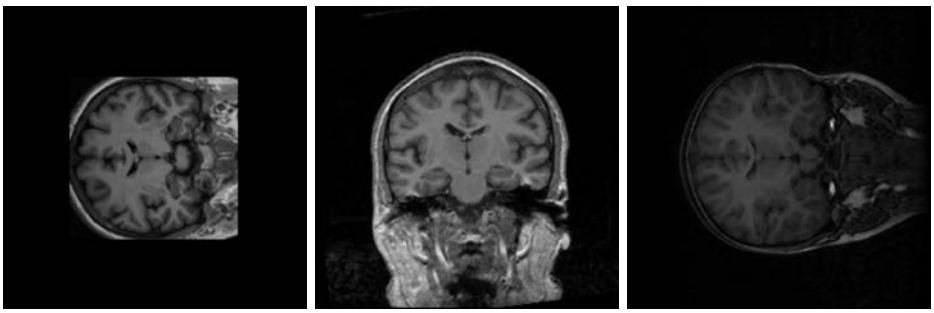
Figure 13.8: Example of presence of hallucinations in restored images for NLM based methods, $\alpha = 4$. Brainweb images are displayed. (a) Original noiseless image, (b) LR input image, (c) MFT3D restored image, (d) NLM based method (BD4M).

non-local means filters (NLM). It can be noted that white spots appear in the superior and right part of the restored image. These spots must not appear since neither the noiseless image nor the LR noisy image evince any imperfections outside the brain. Therefore, they are not real features, but hallucinations. The reason why this happens is that NLM learns from the data, which could be from any similar but far apart region of the image, and may introduce features from other regions of the brain in the wrong place. However, MFT3D only takes into account the voxels which are neighboring to the estimated one so it does not suffer this kind of imperfections, and preserves image details better that non-local filters. Therefore it can be ensured that the brain anatomy will not be affected by hallucinations.

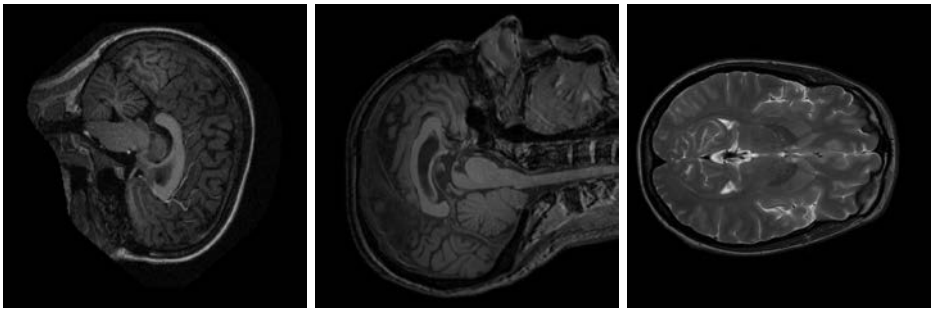
13.3.4 Real data

In order to ensure the repeatability of our experiments, two different public databases of real images were considered: IBSR (Worth, 2010) and OASIS (Marcus et al., 2007). In addition, two images were provided by CIMES, a medical research center of the General Foundation of the University of Málaga (FGUMA) (`http://www.cimes.es/`).

From IBSR we have selected the 1320_2 image (Figure 13.9a). This MRI scan was acquired for a 5 year-old male subject with a 1.5 Tesla General Electric Signa scanner. Contiguous 1.5 mm 3D coronal T1-weighted spoiled gradient echo images of the entire brain were acquired with the following parameters: TR = 40 ms, TE = 5 ms, flip angle = 40°, field of view = 24 cm, matrix = 256×256, and averages = 1. The image size is $256 \times 256 \times 128$ with a normalized voxel size of 1.5 mm (coronal), 0.9375 mm (axial), and 0.9375 mm (sagittal). Its original values are in the range [0,445]. We have also used the 788_6 image from IBSR (Figure 13.9b). This scan was obtained for a 55 year-old male subject with the same parameters as the 1320_2, except for



(a) IBSR 1320_2 (b) IBSR 788_6 (c) IBSR_07



(d) OASIS (e) CIMES T1-weighted (f) CIMES T2-weighted

Figure 13.9: Sample slice of the original real images.

the size, $256 \times 256 \times 60$, and spacing of the pixels: 3.0 mm (coronal), 1.0 mm (axial), and 1.0 mm (sagital). Its original values are in the range [0,255] (8 bit precision). Finally, we have employed the IBSR_07 image (Figure 13.9c). The image size is $256 \times 256 \times 128$, with pixel spacing 1.5 mm (coronal), 1.0 mm (axial), and 1.0 mm (sagittal). The original pixel values are in the range [0,146].

From OASIS, an open access series of imaging studies, we have selected a scan from the cross-sectional MRI data (Figure 13.9d). In particular, we used a T1-weighted magnetization prepared rapid gradient-echo (MP-RAGE) scan from the reliability data set, named OAS1_0080_MR1 and acquired for a 25 year-old female nondemented subject with a 1.5-T Vision scanner (Siemens, Erlangen, Germany). The following parameters were used: TR = 9.7 ms, TE = 4 ms, flip angle = 10°, and matrix = 256×256. The image size is$256 \times 256 \times 128$ with a normalized voxel size of 1.25 mm (coronal), 1.0 mm (axial), and 1.0 mm (sagittal). Its original values are in the range [0,4095] (12 bit precision).

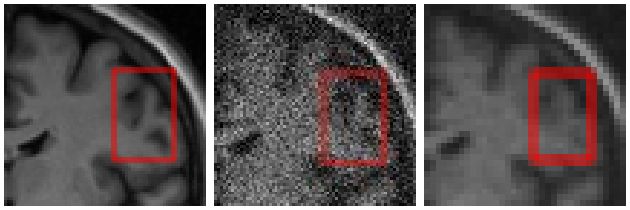
Finally, we have obtained T1-weighted and T2-weighted MR images from CIMES (Figures 13.9e-13.9f). The MRI studies were performed on a 3-T MRI scanner (Philips 3T Intera, Release 3.2.3.2, with an eight-channel platform), with a MASTER gradient system (nominal max gradient strength = 30

mT/m, max slew rate = 150mT/m/ms), equipped with a six-channel Philips SENSE head coil. Head movements were minimized using head pads and a forehead strap. The high-resolution T1-weighted image of the whole brain was acquired for a 50 year-old male using a field of view of 240 mm, TR = 9,9 ms, TE = 4,6 ms, flip angle = 8°, echo train length = 100. Slice thickness was 1.0 mm and 190 slices were acquired, without any gap between slices; acquisition voxels were 0.93 mm x 0.93 mm in-plane, in a 256x256 matrix. The image size is $256 \times 256 \times 190$, with pixel spacing 0.93 mm coronal, 0.93 mm axial, and 1.0 mm sagittal. The original pixel values are in the range [0,4641]. The T2 brain scan was acquired with TR 3000 ms, TE 80 ms, echo train length 15, 40 axially oriented slices, with 3 mm slice thickness and slice gap 0.98 mm. The in-plane reconstructed matrix size was of 512x512, with a voxel size 0.45 mm x 0.45mm. The image size is $512 \times 512 \times 40$ pixels, with pixel spacing 0.45 mm coronal, 0.98 mm axial, and 0.45 mm sagital. The original pixel values are in the range [0,2713].

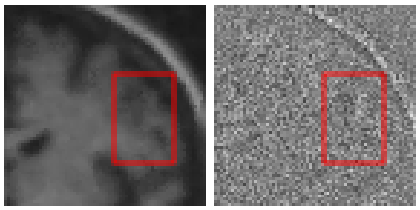
We have only obtained qualitative results for these three datasets, since the original images already have noise. Besides, ideal residual images can not be obtained to compare with the generated from the methods. The noise present in the real images is too low, so as proposed in (Anand and Sahambi, 2010), it has been augmented by simulated 9% Rician noise to achieve a better discrimination between the different compared methods.

Our approach yields good results, specially in terms of sharpness. We can recover brain structures that the other methods lose with large zoom factors. This can be seen in the red rectangles in Figures 13.10, 13.11 and 13.16, where there are very definite surcus and gyrus in the brain of the original images but they can not be appreciated in the competitors' results. In Figures 13.12, 13.14 and 13.15 it can be seen that the restored images obtained by the other methods are oversmoothed, and ours is the only one that can obtain a HR image similar to the original one, even when large zoom factors are employed. That is, MFT3D is able to remove the aliasing while keeping the small features, but all the competitors remove the aliasing artifacts at the expense of oversmoothing. If we compare the residual images, we can see that the competitors manifest accrued losses of brain structures, i.e. anatomical features are visible in the residual image for the competitors. All methods remove noise properly, but NLM based methods remove more in the background, which is useless. Our method provides a good balance between noise removal and detail preservation.

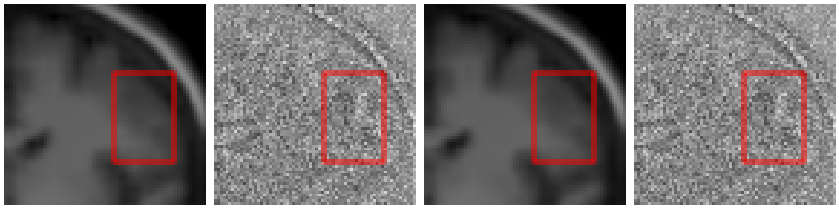
Real data processing by the competing NLM based methods is affected by hallucinations. For example, in Figure 13.13 are displayed the axial slices for the OASIS real image. A zoom factor of $\alpha = 3$ was used and it is displayed the original image, the generated LR, which has been augmented with heavy noise, and the SR results for MFT3D and PRI-NLM3D. Two big spots appear in the left side of the competitor image. As we move along



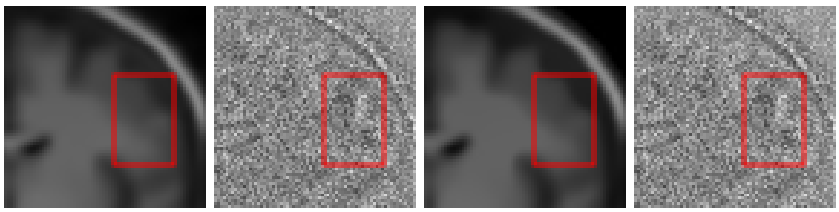
(a) Original HR image (b) 9% Rician noise added (c) LR input image



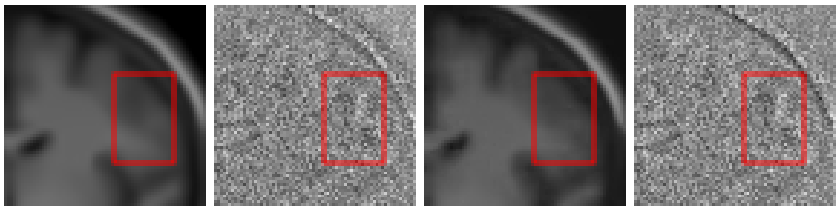
(d) MFT3D result (e) MFT3D residual



(f) NLM3D result (g) NLM3D residual (h) WSM result (i) WSM residual

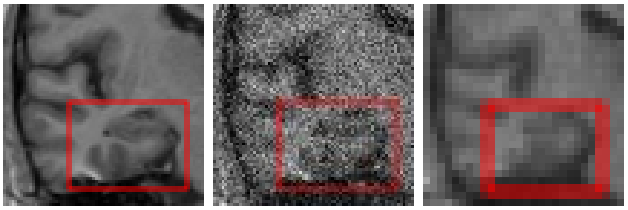


(j) ODCT3D result (k) ODCT3D residual (l) PRI_NLM3D result (m) PRI_NLM3D residual

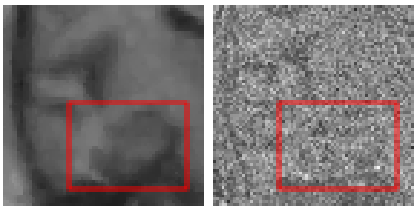


(n) BD4M result (o) BD4M residual (p) NLMU result (q) NLMU residual

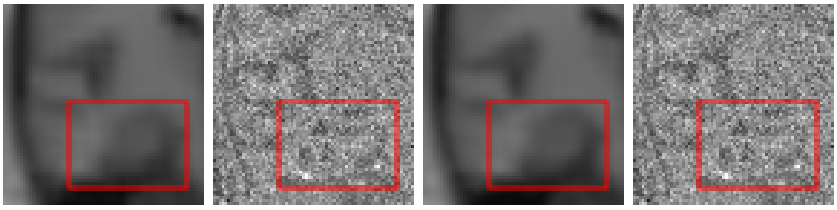
Figure 13.10: Qualitative results for a detail of the IBSR_07 T1 weighted real image, $\alpha = 2$, coronal view.



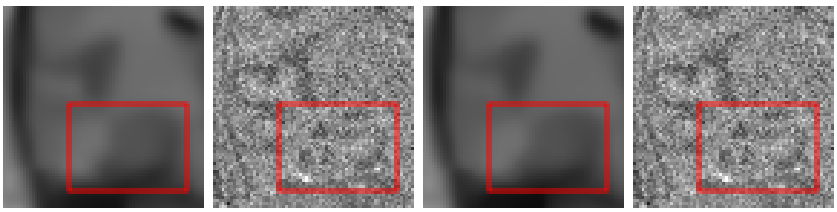
(a) Original HR image (b) 9% Rician noise added (c) LR input image



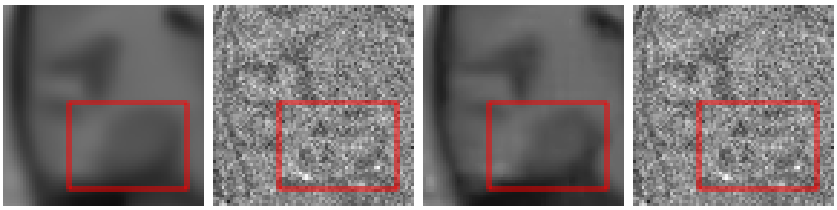
(d) MFT3D result (e) MFT3D residual



(f) NLM3D result (g) NLM3D residual (h) WSM result (i) WSM residual

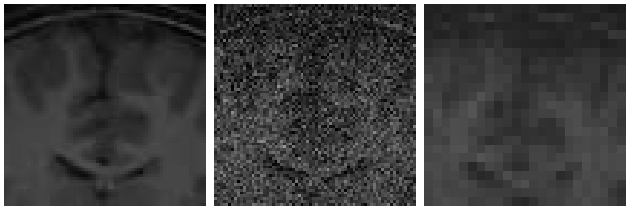


(j) ODCT3D result (k) ODCT3D residual (l) PRI_NLM3D result (m) PRI_NLM3D residual

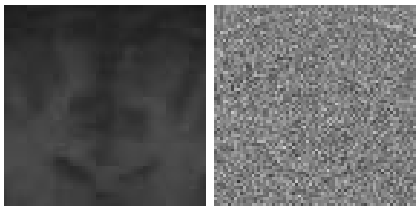


(n) BD4M result (o) BD4M residual (p) NLMU result (q) NLMU residual

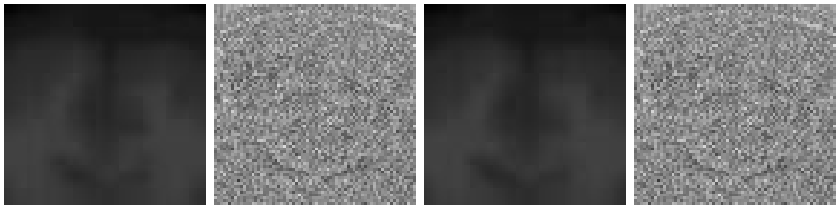
Figure 13.11: Qualitative results for a detail of the 788_6 T1 weighted real image, $\alpha = 2.5$, coronal view.



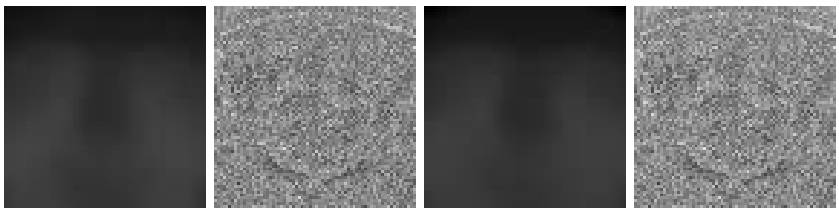
(a) Original HR image (b) 9% Rician noise added (c) LR input image



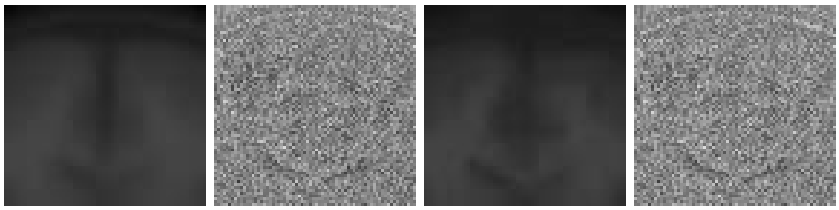
(d) MFT3D result (e) MFT3D residual



(f) NLM3D result (g) NLM3D residual (h) WSM result (i) WSM residual

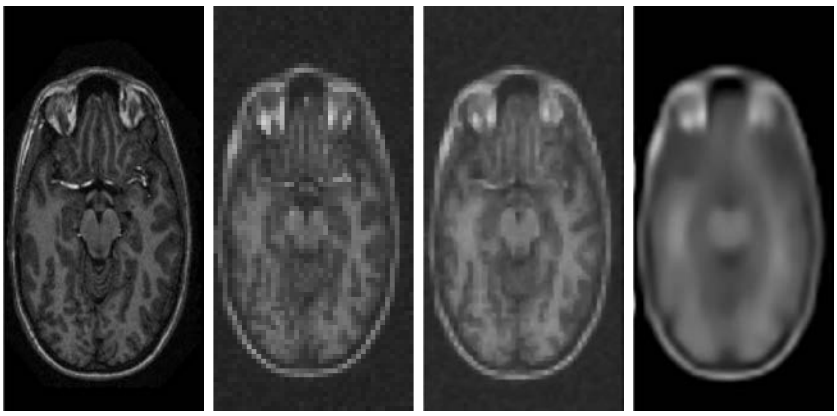


(j) ODCT3D result (k) ODCT3D residual (l) PRI_NLM3D result (m) PRI_NLM3D residual



(n) BD4M result (o) BD4M residual (p) NLMU result (q) NLMU residual

Figure 13.12: Qualitative results for a detail of the 1320_2 T1 weighted real image, $\alpha = 3$, coronal view.



(a) (b) (c) (d)

Figure 13.13: Example of presence of hallucinations in restored images for NLM based methods, $\alpha = 3$. OASIS images are displayed. (a) Original noiseless image, (b) LR input image, (c) MFT3D restored image, (d) NLM based method (PRI-NLM3D).

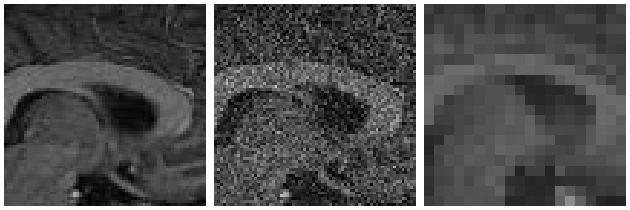
the axial slices, these spots still remain and they move up and down in the image. Neither in the original, LR or our approach image we can see similar distortions. Again, this shows the robustness of the MFT3D method for any kind of input image.

In which respects to the CPU time employed, MFT3D is slower than all the other methods. On average, they need around 2 minutes at most and our method need at least 1 hour to compute the restored image, depending on the size factor employed for the SR. The utilization of a GPU would improve between 25 and 50 times the processing time, since each of the H tilings of the MFT3D can be computed in parallel.

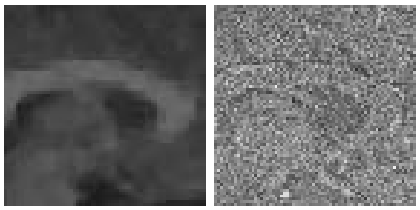
13.4 Conclusions

We have presented a new method for both super-resolution and noise removal applied to 3D magnetic resonance images. It is based on the application of median filters in two steps. First, it is applied over random parallelepipeds tiling the space repeatedly to obtain an initial HR image, and then the final image is obtained by computing the median of each repetition. Since the median is used for denoising, the noise distribution in the image is not relevant, and could be unknown. The convergence of the method has been studied from a mathematical point of view.

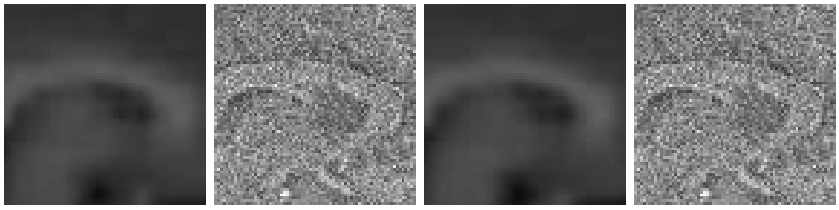
Experimental results have been carried out with both synthetic and real images. Different types of images (T1, T2 and PD), and various noise levels and zoom factors have been tested. The proposed method yields better



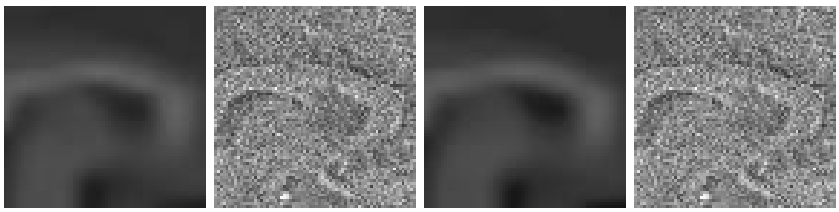
(a) Original HR image (b) 9% Rician noise added (c) LR input image



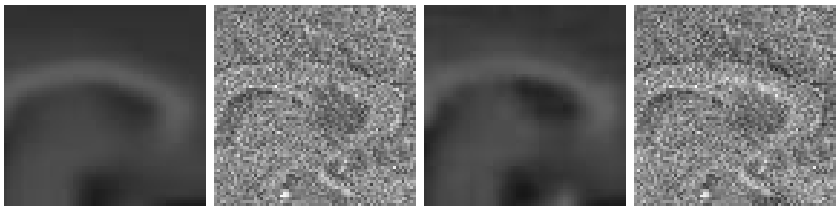
(d) MFT3D result (e) MFT3D residual



(f) NLM3D result (g) NLM3D residual (h) WSM result (i) WSM residual

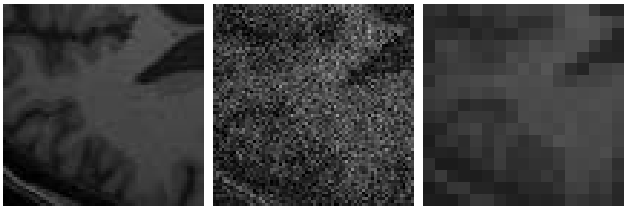


(j) ODCT3D result (k) ODCT3D residual (l) PRI_NLM3D result (m) PRI_NLM3D residual

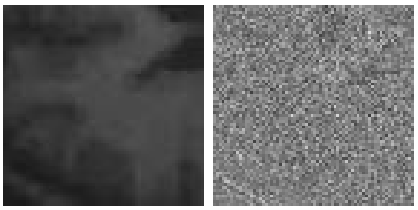


(n) BD4M result (o) BD4M residual (p) NLMU result (q) NLMU residual

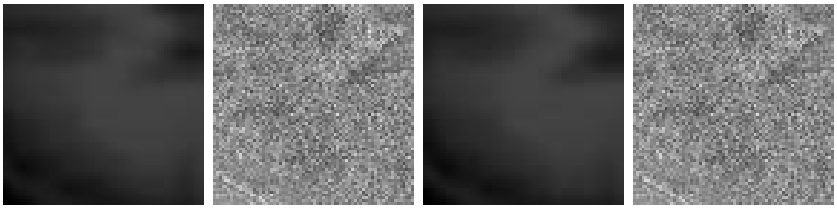
Figure 13.14: Qualitative results for a detail of the T1 weighted real image from OASIS, $\alpha = 3.5$, sagital view.



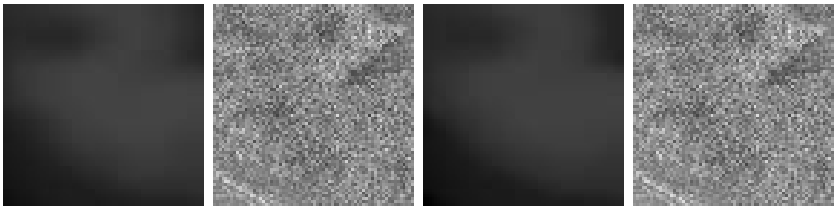
(a) Original HR image (b) 9% Rician noise added (c) LR input image



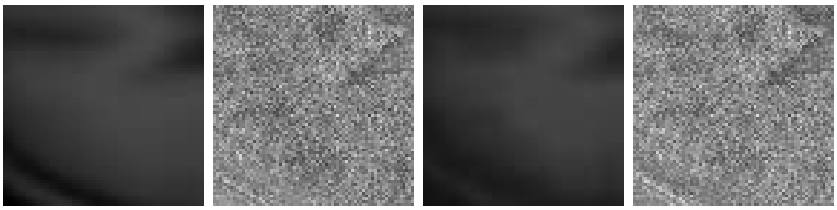
(d) MFT3D result (e) MFT3D residual



(f) NLM3D result (g) NLM3D residual (h) WSM result (i) WSM residual

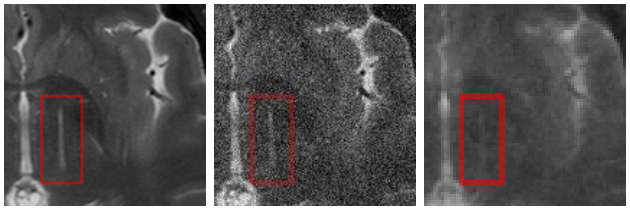


(j) ODCT3D result (k) ODCT3D residual (l) PRI_NLM3D result (m) PRI_NLM3D residual

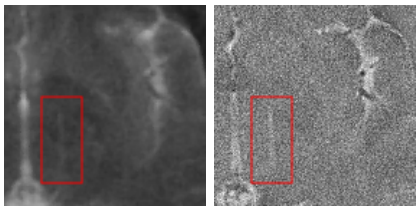


(n) BD4M result (o) BD4M residual (p) NLMU result (q) NLMU residual

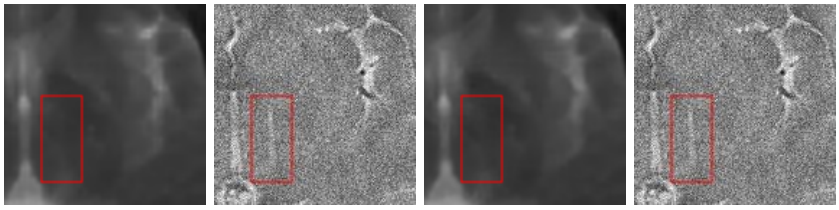
Figure 13.15: Qualitative results for a detail of the T1 weighted real image from CIMES, $\alpha = 4$, axial view.



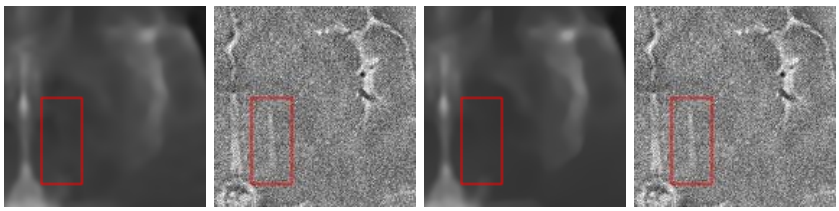
(a) Original HR image (b) 9% Rician noise added (c) LR input image



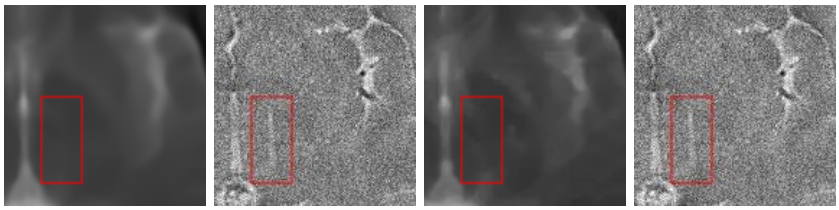
(d) MFT3D result (e) MFT3D residual



(f) NLM3D result (g) NLM3D residual (h) WSM result (i) WSM residual



(j) ODCT3D result (k) ODCT3D residual (l) PRI_NLM3D result (m) PRI_NLM3D residual



(n) BD4M result (o) BD4M residual (p) NLMU result (q) NLMU residual

Figure 13.16: Qualitative results for a detail of the T2 weighted real image from CIMES, $\alpha = 2$, axial view.

quantitative results than other state-of-the-art methods when 5% or more Rician noise is present in the images, irrespective of the scale factor employed in the super-resolution process. Any integer or fractional zoom factor can be employed, but it has been necessary to use an interpolation technique for the competing methods because they are limited to a small range of values.

It has been shown that MFT3D also achieves good qualitative results when heavy noise is present in the image. The noise is removed while the finer details of the image are preserved. In contrast to this, the competing methods suffer from oversmoothing and relevant anatomical features are lost. This can incur in errors since a lesion can disappear due to its similarity to noisy pixels. The low values of MSE and high values of SSIM confirm the better quality of the HR images obtained by our proposal. Moreover, NLM based methods generate hallucinations in the HR image, which could be largely detrimental for medical diagnosis, while our approach does not generate any hallucinations.

Chapter 14

Deep Learning networks with ℓ_p-norm loss layers for spatial resolution enhancement of 3D medical images

Computer science is to biology what calculus is to physics. It's the natural mathematical technique that best maps the character of the subject.

Harold Morowitz

Abstract: Nowadays, obtaining high-quality Magnetic Resonance (MR) images is a complex problem due to several acquisition factors, but is crucial in order to perform good diagnostics. The enhancement of the resolution is a typical procedure applied after the image generation. State-of-art works gather a large variety of methods for Super-Resolution (SR), among which deep learning has become very popular during the last years. Most of the SR deep-learning methods are based on the minimization of the residuals by the use of Euclidean loss layers. In this research, we propose an SR model based on the use of a ℓ_p-norm loss layer to improve the learning process and obtain a better High-Resolution (HR) image. This method was implemented using a three-dimensional CNN, and tested for several norms in order to determine the most robust fit. The proposed methodology was trained and tested with sets of MR structural T1-weighted images and showed better outcomes quantitatively, in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), and the restored and the calculated residual images showed better CNN outputs.

14.1 Introduction

Many different approaches have been applied to obtain resolution-enhanced images in the recent years, and, among them, deep learning has gained increasing popularity. Most of these methods are based on the minimization of the residuals using Euclidean loss layers.

In recent times, the use of ℓ_p-norm methods in optimization algorithms has drawn attention for various applications. They have been used for machine learning binary classifiers (Grove et al., 2001; Gentile, 2003), and in the framework of optimal control (Blueschke and Savin, 2017). Due to the properties of the ℓ_p-norm when dealing with with sparse vectors and matrices (Chen et al., 2010; Li et al., 2018b; Zhang et al., 2013; Abramovich et al., 2006), it has also been applied for feature selection (Ye et al., 2017).

One of the advantages of using a ℓ_p-norm, with $1 < p < 2$, is that it can allow reducing the effect of outliers in a minimization problem. Noise and artifacts in the images of the training set for MR superresolution are outliers that have to be avoided so that a SR algorithm provides realistic high quality solutions. In this work, the ℓ_p-norm with $1 < p < 2$ is proposed as the loss function for neural layers of super resolution convolutional neural networks, and the most suitable values of p are studied.

The rest of chapter is organized as follows: Section 14.2 contains the theoretical background of our model. Then, in Section 14.3 the experiments carried out and the outcomes are described. Finally, the conclusions and future works are presented in Section 14.4.

14.2 The model

In this section we present the learning rule for neural layers with ℓ_p-norm loss function. The rationale behind our proposal is that the classic quadratic loss function, which corresponds to $p = 2$, may be outperformed by loss functions based on the p-th power of the absolute value of the error. For $p < 2$, this increases the robustness of the learning rule against outliers, i.e. training samples with extremely large values of the error.

The ℓ_p-norm of a D-dimensional vector $\mathbf{v} \in \mathbb{R}^D$ can be defined as:

$$\|\mathbf{v}\|_p = \sum_{j=1}^{D} |v_j|^p \tag{14.1}$$

Without loss of generality, the exponent $1/p$ defined in (5.11) can be omitted.

For deep learning neural networks, the standard loss function is the square of the Euclidean (ℓ_2-norm) of the difference between the desired output vector and the output vector obtained from the network, averaged for all available training samples. While this choice yields excellent results for

many applications, there is room for improvement, since the exponent p in (14.1) determines the importance that the loss function gives to those training samples which have components with higher absolute values $|v_j|$. The higher p, the more importance that is given to extreme values of $|v_j|$. It must be considered that maybe those extreme values of $|v_j|$ correspond to badly measured training samples, or irrelevant observations. Therefore, a loss function which does not get too influenced by those extreme errors might obtain better values than the ℓ_2-norm. The most promising values of p are those for which the ℓ_p-norm fulfils the mathematical definition of norm, i.e. $1 \leq p \leq 2$.

Given the above considerations, we propose to employ neural layers with a ℓ_p-norm loss function. The loss function is defined as follows, for N samples of dimension D:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{D} |y_{ij} - z_{ij}|^p \tag{14.2}$$

where y_{ij} is the j-th component of the i-th obtained output vector from the network and z_{ij} is the desired output.

The derivative of the loss function with respect to a synaptic weight w is:

$$\frac{\partial E}{\partial w} = \sum_{i=1}^{N} \sum_{j=1}^{D} p \, |y_{ij} - z_{ij}|^{p-1} \operatorname{sign}(y_{ij} - z_{ij}) \frac{\partial y_{ij}}{\partial w} \tag{14.3}$$

where the sign function is defined as follows:

$$\operatorname{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{14.4}$$

Stochastic gradient methods can be applied to the gradient of the loss function (14.3), in order to train a ℓ_p-norm neural layer within a deep neural network.

14.3 Experimental results

This section describes the experiments we carried on. First, Subsection 14.3.2 explains the low-resolution image generation, as well as the software and hardware employed, and the selected performance metrics for comparison between ℓ_p-norms. The input datasets are described in Subsection 14.3.1. Finally, we report the findings of the experiments in Subsection 14.3.3.

14.3.1 Datasets

Six different T1-weighted MR images are considered for the evaluation of the ℓ_p-norms:

- 2 images of the Kirby 21 (images 10 and 11) (Landman et al., 2011). These data were acquired using a 3-T MR scanner with a $1.0 \times 1.0 \times 1.2\,mm^3$ voxel resolution over an field-of-view (FOV) of $240 \times 204 \times 256\,mm$ acquired in the sagittal plane.
- 2 images of the OASIS dataset (images 1 and 2 of the cross-sectional data) (Marcus et al., 2007) . Data were acquired on a 1.5-T Vision scanner with a $1.0 \times 1.0 \times 1.25\,mm^3$ voxel resolution over an FOV of $256 \times 256\,mm$.
- 1 image of the IBSR public dataset (Worth, 2010). It is named IBSR_07, it has image size $256 \times 256 \times 128$, with $1.5 \times 1.0 \times 1.0\,mm^3$ voxel resolution.
- 1 T1-weighted image from CIMES[1] using a 3-T MR scanner with a $0.93 \times 0.93 \times 1.0\,mm^3$ voxel resolution over an FOV of $256 \times 256\,mm$.

14.3.2 Methods

The deep network used to evaluate our proposal was the SRCNN3D method (Pham et al., 2017), which is a super-resolution convolutional neural network for three-dimensional MR images. The convolutional network have been developed using Caffe package (Jia et al., 2014) on a Python framework. One of the motivations of selecting this network is its simplicity to understand and modify the source code to create our customized ℓ_p-norm loss layer.

Given an image $\mathbf{Y}$ and its respective LR one $\mathbf{X}$, this CNN is based on the application of three blocks of convolutional Rectified Linear Unit (ReLU) layers successively, to a pre-interpolated image $\mathbf{Z} = I(\mathbf{X})$. This step is internal to the method. Thus, the net computes a super-resoluted HR image by the minimization of the Euclidean loss between the output of the CNN, $s(\mathbf{Z})$, and the original HR image $\mathbf{Y}$.

$$f = arg\,min_s \sum ||\mathbf{Y} - s(\mathbf{Z})||^2 \tag{14.5}$$

This network is trained using overlapping patches that are extracted from a set of HR reference images. A down-sampling and up-sampling is applied to each patch and a set of pairs input-target is created in order to learn an end-to-end function between low and high resolution images. Specific details of the implementation of this network can be found in the literature.

We carried out a training over 50000 iterations for each ℓ_p-norm, using momentum of 0.9, learning rate of 0.0001 and batch size of 256. Stochastic Gradient Descent (SGD) was used for model optimization (all are default parameters). Images 33-42 from Kirby dataset were used for training. Furthermore, zoom factors 2 and 3 were employed in our analysis. Figure 14.1

[1] https://fguma.es/unidad-imagen-molecular/

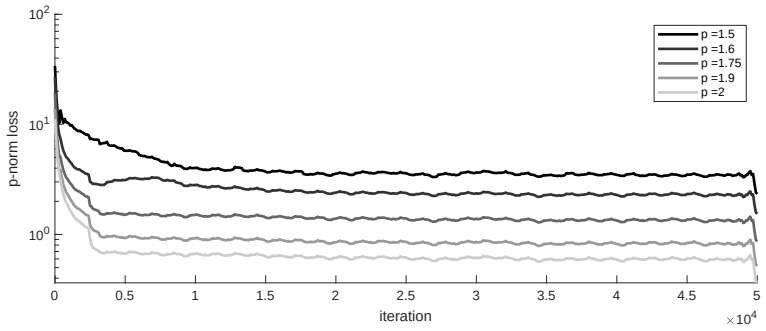


Figure 14.1: Training loss for the ℓ_p-norms ($p \in \{1.5, 1.6, 1.75, 1.9, 2\}$) during 50000 iterations, smoothed using a sliding window of 50.

shows the training loss curves for each value of p. We used a fixed number of iterations for every training to make a fair comparison, and the selected number of iterations assess the convergence of all the trainings. We have also tested $1 \leq p < 1.5$ but no convergence was achieved. The comparison experiments have been carried out on a 64-bit Personal Computer with an six-core Intel i7 3.50GHz CPU, 64 GB RAM, with a GPU Nvidia GTX Titan.

On the other hand, LR images were created applying the following algorithm to the HR images:

1. Crop HR image dimensions to make divisible by the scale factor.
2. Apply a 3D Gaussian filter with standard deviation equal to 1.
3. Use an bi-cubic interpolation method to generate LR image

In this work, MATLAB fuctions were used with default parameters.

Three quality measures were used to evaluate the proposed method:

- *Peak Signal-to-Noise Ratio (PSNR)*, measured in (decibels) dB, and the more high value, the better is the likeness.

$$PSNR = 10 \log_{10} \left(\frac{peakval^2}{MSE} \right) \tag{14.6}$$

  where *peakval* is maximum possible value of the image and MSE refers to the Mean Squared Error.

- *Structural Similarity Index Measure (SSIM)* (Wang et al., 2004), which focuses on structural similarities between images, returning a value between 0 and 1 (higher is better):

$$SSIM(x, y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{14.7}$$

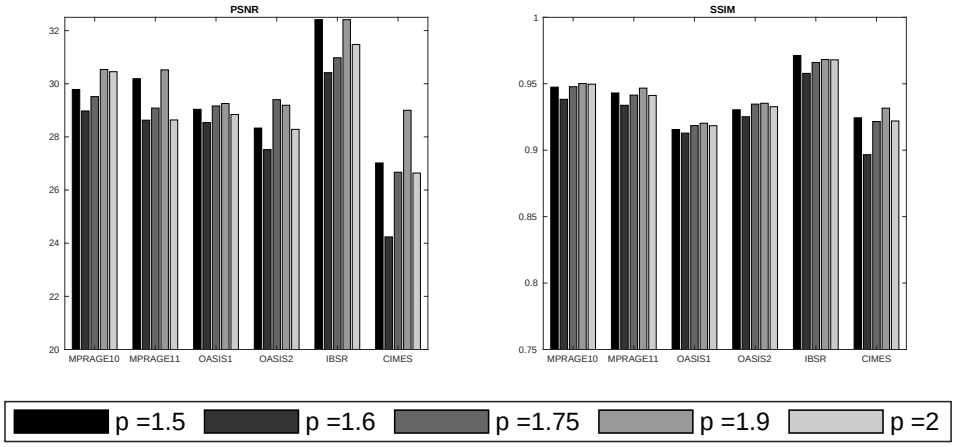


Figure 14.2: Comparison of the PSNR and SSIM for the ℓ_p-norms ($p \in \{1.5, 1.6, 1.75, 1.9, 2\}$) and the six tested images, using a scale factor equal to 2.

where $\mu_x$and μ_y are the mean value of images x and y, σ_x and σ_y are the standard deviation of images x and y, σ_{xy} is the covariance of x and y, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ (default values were used: $L = 1$ is the dynamic range, $k_1 = 0.01$ and $k_2 = 0.03$).

Besides, we used residual images to analyze the results from a qualitative point of view:

$$residual = h - s \tag{14.8}$$

where h represents the original HR image and s the output of the CNN. The best performance is such that the residual image is the zero matrix. As it is difficult to distinguish dark values, we subtracted the constant 0.5 to the residual images in order to see the performance differences. Thus, the residual images appear in gray.

14.3.3 Results

First, we evaluate each convolutional neural network from a quantitative point of view. Performance results in terms of PSNR and SSIM for each image applying a zoom factor of 2, are depicted in Figure 14.2. Each color bar represents one value of p. The higher is the bar, the better is the network. In both metrics, for five of the six tested images the "winner" is the $\ell_{1.9}$-norm.

If we focus on the difference with respect to the euclidean norm, there are great improvements for MPRAGE11 and CIMES images. $p = 1.5$ also shows good performance in PSNR compared to $p = 2$, so it is shown that it is not always the best and we can get better results with a different minimization. However, there are irregularities for $p = 1.6$ and $p = 1.75$, as for some images

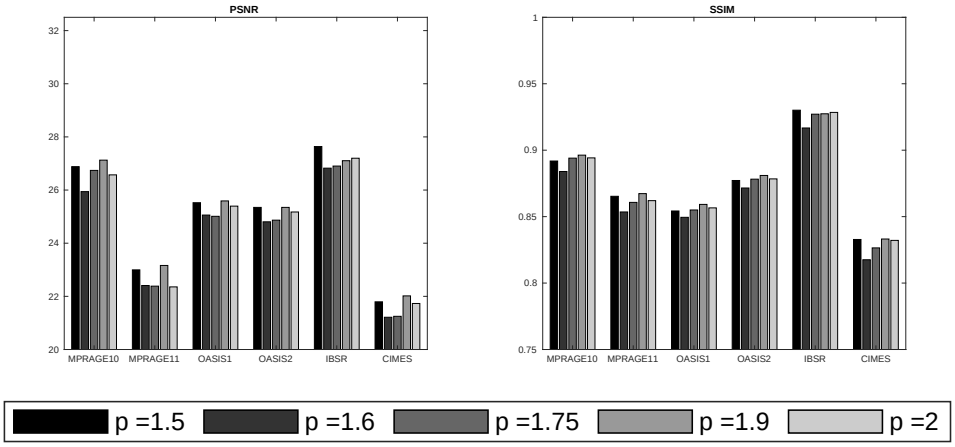
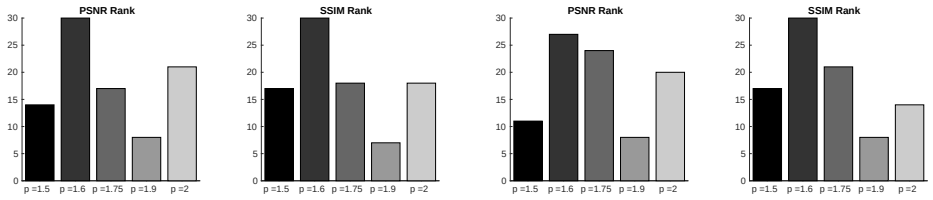


Figure 14.3: Comparison of the PSNR and SSIM for the ℓ_p-norms ($p \in \{1.5, 1.6, 1.75, 1.9, 2\}$) and the six tested images, using a scale factor equal to 3.



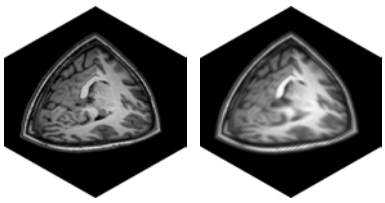
(a) Zoom 2 (b) Zoom 3

Figure 14.4: Ranking of the PSNR and SSIM for the ℓ_p-norms for scale factors 2 and 3, taking into account all the images.

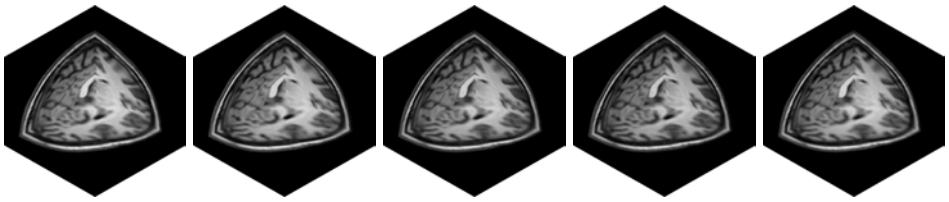
performs better but for others worse. This analysis can be extrapolated to the SSIM metric too, where the differences between nets are smaller but similar.

Figure 14.3 shows the results for zoom factor 3. In general, the performance is worse because we are trying to infer more information from the LR image, and and difference between images is greater. Here again the pattern is the same. The networks based on the $\ell_{1.5}$-norm and the $\ell_{1.9}$-norm carried out a better prediction of most of the images, being $p = 1.9$ the best one in both PSNR and SSIM.

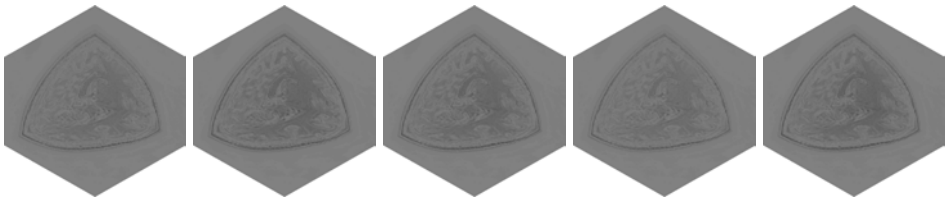
The general behaviour of each neural network across all images is summarized in Figure 14.4. We computed a ranking sorting each ℓ_p-norm according to its performance with respect each image and we assigned points from 1 to 5. The less punctuation the better is the network. For scale 2, shown in Figure 14.4a, the best method is that one based on the $\ell_{1.9}$-norm loss layer.



(a) Original HR image (b) LR image



(c) $p = 1.5$ (d) $p = 1.6$ (e) $p = 1.75$ (f) $p = 1.9$ (g) $p = 2$



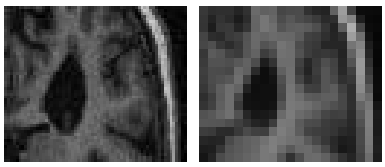
(h) $p = 1.5$ (i) $p = 1.6$ (j) $p = 1.75$ (k) $p = 1.9$ (l) $p = 2$

Figure 14.5: Qualitative results for $MPRAGE11$ image for each ℓ_p-norm, applied with zoom factor 2. Second row shows the reconstructed image by each algorithm and third row shows residual images between the reconstructed and the original HR image.

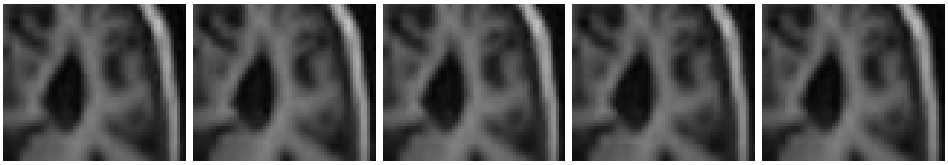
The performance of $p = 1.6$ in terms of both PSNR and SSIM is clearly the worst, followed by the ℓ_2-norm, specially in PSNR. For the similarity metric there are not great differences between 2, 1.5 and 1.75. Rank of zoom 3 is depicted in Figure 14.4b. Here the differences with respect to zoom 2 go in favour of $p = 2$, which is situated in the second place of our SSIM rank. Nevertheless, $p = 1.9$ is still the best method, followed closely by $p = 1.5$ in PSNR.

On the other hand, we compare the output of the networks from a qualitative point of view in Figure 14.5 and 14.6. Residual images are also displayed to have a better discrimination between methods.

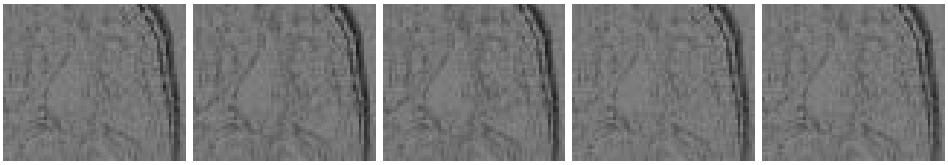
Firstly, in Figure 14.5 is shown a three-dimensional perspective of the CNN's outcome of image 11 of the Kirby dataset. The differences can be seen on the third row, where the intensity of gray varies from one ℓ_p-norm to other. The darkest images are the ones corresponding to $p = 1.6$ and $p = 2$, which indicates that the difference between the output of the net



(a) Original HR image (b) LR image



(c) $p = 1.5$ (d) $p = 1.6$ (e) $p = 1.75$ (f) $p = 1.9$ (g) $p = 2$



(h) $p = 1.5$ (i) $p = 1.6$ (j) $p = 1.75$ (k) $p = 1.9$ (l) $p = 2$

Figure 14.6: Qualitative results for a section of the axial view of the $OASIS1$ image for each ℓ_p-norm, applied with zoom factor 3. Second row shows the reconstructed image by each algorithm and third row shows residual images between the reconstructed and the original HR image.

and the ground truth is greater. On the other hand, the image where less structures are removed, that is, the most gray uniform one, corresponds to $p = 1.9$. This results matches with the previous quantitative analysis, where its values of PSNR and SSIM are the best.

An axial section of the image 1 of the OASIS dataset is compared in Figure 14.6. In this case, a zoom factor of 3 is used. The main differences can be seen in the borders of the lateral ventricle (the uniform centered region). The output with less dark border is the one produced using $p = 1.9$, followed by $p = 2$ and $p = 1.5$.This can indicates that the structural surcus of the cerebrum are restored in a proper way. Moreover, some dark spots are conserved better since they appear clearer in the residual image of our proposed method, which is fundamental to not carry on a bad diagnostic.

14.4 Conclusions

This work presents a robust three-dimensional super-resolution method for magnetic resonance images. It is based on the use of a ℓ_p-norm loss layer

instead of the usual Euclidean formulation. High-resolution images are obtained by learning a map between an interpolated low-resolution image and the ground truth, where the optimization function is defined as the ℓ_p-norm to reduce the overall error in the training. T1 structural images from different origins were used to evaluate the efficiency of each variant of the network, achieving better performance. Results show that the Euclidean loss layer is not always the best norm, since PSNR and SSIM have increased for most images and zoom factors. Qualitatively, restored images look more refined and with less structural degradation. The proposed approach could be extended to another neural networks in order to improve the quality of the outputs.

Chapter 15

Multiobjective optimization of deep neural networks with combinations of ℓ_p-norm cost functions for 3D medical SR

In theory there is no difference between theory and practice. In practice there is.

Yogi Berra

Abstract: In medical imaging, the lack of high-quality images is present in many areas such as Magnetic Resonance (MR). Due to many acquisition impediments, the generated images have not enough resolution to carry out an adequate diagnosis. Image Super-Resolution (SR) is an ill-posed problem that tries to infer information from the image to enhance its resolution. Nowadays, deep learning techniques have become a powerful tool to extract features from images and infer new information. In MR, most of the recent works are based on the minimization of the errors between the input and the output images based on the Euclidean norm. This work presents a new methodology to perform three-dimensional SR based on the combination of ℓ_p-norms in the loss layer. Two multiobjective optimization techniques are used to combine two cost functions. The proposed loss layers were trained with the SRCNN3D and DCSRN networks and tested with two MR structural T1-weighted datasets, and then compared with the traditional Euclidean loss. Experimental results show significant differences in terms of Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and Bhattacharyya Coefficient (BC), while the residual images show refined details.

15.1 Introduction

Deep learning neural networks use loss functions that are commonly based on the squared Euclidean norm. However, the use of alternatives like ℓ_p-norms has attracted attention for a variety of tasks. The ℓ_p-norm is analogous to the Euclidean norm where the exponent 2 is substituted by an alternative value p. The robustness properties of the ℓ_p-norm have been extensively studied, as seen in (Chen et al., 2017b; Gao, 2001) and references therein. Standard minimization of the Euclidean norm is known to be optimal for Gaussian noise, while ℓ_p-norms can perform better for datasets containing outliers and non-Gaussian noise (Chen et al., 2016). Typical machine learning applications such as binary classification have already taken advantage of ℓ_p-norm techniques (Grove et al., 2001; Gentile, 2003). Optimal control has also employed this kind of norms (Blueschke and Savin, 2017). The favorable properties of the p-norm to manage sparsity in matrices and vectors have been widely recognized (Chen et al., 2010; Li et al., 2018b; Zhang et al., 2013; Abramovich et al., 2006), which has led to applications in the feature selection field (Ye et al., 2017).

Using an ℓ_p-norm approach can be useful to reduce the effect of outliers in minimization problems when $p < 2$ values are selected. If p lies within the $[1, 2]$ interval it will also fulfill the formal definition of a geometrical norm. In MRI applications, using p within this range can allow compensating for noise and artifacts that are often present in the images, thereby increasing the robustness of an SR algorithm. We start out, then, proposing to improve a deep learning neural network by using p values in the $[1, 2]$ interval.

However, it is still interesting to be able to count on the robustness of the traditional squared Euclidean norm. Multiobjective optimization methods have been developed for the case where more than one goal is aimed at in the optimization process. For example, image enhancement aims to reduce the noise while preserving the small details. Usually, these two goals clash, since noise reduction is often achieved by smoothing out the image and thereby removing the details. Multiobjective optimization makes it possible to combine two or more loss functions and, depending on the methods and parameters used, to give more relevance to the most relevant goal in each case. A fundamental concept related to multiobjective optimization is that of the Pareto front (Emmerich and Deutz, 2018), which is a surface in the space of possible solutions which comprises all solutions that are not dominated by any other solution, i.e. there is no solution which is equal or better than them for all the goals which are optimized.

Multiobjective optimization is based on the premise that the improvement of one of the objectives may lead to the deterioration of another objective. Therefore, a globally optimal solution is not possible, so that a search over the Pareto front is required. Evolutionary algorithms can be

employed to approximate the Pareto front of a problem within a population of possible solutions, which has led to their extensive application (Zhou et al., 2011). Popular current proposals for multiobjective optimization include heuristic methods based on evolutionary algorithms (Su et al., 2019). Among these, genetic algorithms, decomposition-based proposals, particle swarm optimization, bat algorithms, harmony search, ant colony optimization, and non-dominated sorting genetic algorithms are designed to cope with various challenges of multiobjective optimization problems. All of them use operators inspired in biological evolution in order to improve a population of possible solutions to the optimization problem. Also, it is very common that several methodologies are hybridized, by combining search and updating methods, or alternating methods in different phases (Zhou et al., 2011). In our case, the number of objectives to be optimized is two, although there are methods to optimize more than four objectives, which is known as many-objective optimization (Li et al., 2020).

Scalarization is another approach to multiobjective optimization, which is based on the optimization of a scalar function which combines the multiple objective functions of the original problem. There are many scalarization methods in multiobjective optimization with different characteristics such as convexity, boundedness, the ability to generate proper efficient solutions, the number of additional constraints, etc. For example, the elastic constraint method (Ehrgott and Ryan, 2002) gives conditions on the characterization for properly efficient solutions, and the augmented weighted Chebyshev scalar problem (Steuer and Choo, 1983) generates properly efficient solutions for certain selected values of weights and augmentation parameter. This work focuses on a particularization of the Pascoletti-Serafini scalarization (Eichfelder, 2009), which guarantees to generate proper solutions just by selecting the proper weights.

In light of the above, our proposal involves combining two different cost functions with different values of p, which should lead to benefiting from the advantages of each of the two configurations. Therefore, the aim of this work is to obtain an improved deep learning network by using a multiobjective optimization approach with two ℓ_p-norm-based loss functions and to find the parameters of the new cost function that yield the best results for their application to three-dimensional MR images. The combination of two alternative cost functions is a novel approach to the training of deep convolutional neural networks since the standard approach to deep learning relies on the minimization by stochastic gradient descent of a single loss function. While the standard approach has obtained remarkable results, our aim is to further improve the performance of the deep networks by making them pursue complementary goals which may help in solving the problem at hand, in our case the super-resolution of a three-dimensional MRI.

The rest of this chapter is organized as follows: The theoretical back-

ground of this work is detailed in Section 15.2. Section 15.3 provides a description of the SR network to be improved, the datasets for testing and the optimization experiments, followed by the obtained results. A discussion of the obtained results is carried out in Section 15.4. Finally, the conclusions and proposals for future work are presented in Section 15.5.

15.2 Methodology

In this section, we propose a new optimization framework to train deep neural networks. Our proposal considers cost functions based on the L_p-norm (Datar et al., 2002; Han et al., 2007) of the error vector. Depending on the specific norm, the optimization goal varies. Therefore, we employ multiobjective optimization techniques (Emmerich and Deutz, 2018; Giagkiozis and Fleming, 2015) in order to combine two cost functions. To do this, scalarization techniques are advocated, since the training process requires a single scalar quantity to be minimized. Subsection 15.2.1 deals with the L_p-norm cost functions, while Subsection 15.2.2 introduces multiobjective optimization of two cost functions by scalarization.

15.2.1 Lp-norm loss functions

Next, the usage of L_p-norm loss functions for deep neural networks is investigated. The L_p-norm has previously been considered as a cost function for signal processing (Bar-Yossef et al., 2004; Kerahroodi et al., 2017; Kuruoğlu et al., 1997; Zhang et al., 2018), image processing (Bioucas-Dias and Valadão, 2007; Unser et al., 1992) and machine learning (Hathaway et al., 2000; Park and Kwak, 2018) tasks. The L_p-norm of a D-dimensional vector $\mathbf{z} \in \mathbb{R}^D$ is given by:

$$\|\mathbf{z}\|_p = \sum_{j=1}^{D} |z_j|^p \tag{15.1}$$

The starting point for our strategy is the realization that the squared Euclidean norm loss function, i.e. $p = 2$, is not the only choice, and it might yield worse results than other loss functions associated to the L_p-norm of the error vector at a certain neural layer. Training samples with unusually high values of the components of the error vector will have a less dominating effect on the optimization if $p < 2$, which means that these values of p are likely to enhance the behavior of the learning rule whenever those extreme values of the error are present. This happens because the higher the value of the exponent p, the faster the L_p-norm grows for increasing values of its argument. For example, if $p = 1$, doubling the argument of the norm doubles the value of the L_p-norm, while if $p = 2$, doubling the argument of the norm quadruples the value of the L_p-norm. Consequently, the effect of outliers

with unusually high values of the error, i.e. high values of the argument of the L_p-norm, is more exacerbated for high values of p.

The standard configuration of a deep learning neural network includes a loss function given by the average for all training data of the squared Euclidean norm ($p = 2$) of the error vector computed from the desired output vector and the actual output vector yielded by the neural layer of interest. The introduction of the L_p-norm means that there is an extra tunable parameter of the learning algorithm, namely the exponent p in (15.1), which controls how much the learning is focused on reducing the error for the training data associated error vectors have components with large absolute values $|z_j|$. Lower values of p mean that less relevance is provided to the training data with the highest values of $|z_j|$. This is advantageous in those situations where the largest errors are associated with measurement errors, impulse noise, missing data, and similar factors. In this context, cost functions other than the squared Euclidean norm might provide better results, given their resilience to the kind of inconveniences mentioned before. In this investigation, the search for good performing exponents p is restricted to the interval of real numbers which yield a L_p-norm which agrees with the formal definition of a geometrical norm, i.e. $p \in [1, 2]$.

In light of the above, we advocate the use of L_p-norm loss functions for the neural layers of a deep neural network. The general definition of a L_p-norm loss function is given by:

$$E_p = \sum_{i=1}^{N} \sum_{j=1}^{D} |y_{ij} - v_{ij}|^p \tag{15.2}$$

where D is the dimension of the samples, N is the number of samples, y_{ij} is the j-th component of the i-th actual output vector and v_{ij} is the desired output vector.

The gradient of the L_p-norm loss (15.2) with respect to a neural weight w is:

$$\frac{\partial E_p}{\partial w} = \sum_{i=1}^{N} \sum_{j=1}^{D} p\, |y_{ij} - v_{ij}|^{p-1} \operatorname{sign}(y_{ij} - v_{ij}) \frac{\partial y_{ij}}{\partial w} \tag{15.3}$$

where

$$\operatorname{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{15.4}$$

15.2.2 Multiobjective optimization by scalarization

The optimization of the L_p-norm loss function (15.2) can lead to different configurations of a network depending on the value of p. This suggests that a combination of two different loss functions E_p and E_q, with $p \neq q$, might merge the advantages of the optimal configurations associated with E_p and

E_q. In other words, multiobjective optimization is proposed for the training of neural layers within a deep neural network. More than two loss functions might be combined, but we have not done it in this work.

As done in standard learning procedures, in our proposal the adjustment of the weights of a neural network is carried out by gradient descent. Since gradient descent minimizes a single loss function, this implies that a combined loss function S must be defined for each neural layer, which integrates the loss functions E_p and E_q. Our choice is driven by the excellent performance of the stochastic gradient descent optimization methods for deep neural networks which are readily available. In multiobjective optimization terms, this means that a scalarization must be performed (Kasimbeyli et al., 2019), although metaheuristics methods could also be considered in future developments of our proposal (Cheng et al., 2015). Two scalarization strategies have been considered in this work:

- Weighted Sum Scalarization (WSS) (Gass and Saaty, 1955). In this simple approach, the combined loss function is defined as the weighted average of the two loss functions:

$$S_{WSS} = \lambda_1 E_p + \lambda_2 E_q \tag{15.5}$$

  where $\lambda_1, \lambda_2 \geq 0$.

- Weighted Chebyshev Scalarization (WCS) (Gong, 2011; Bowman, 1976). This strategy is based on the previous calculation of an ideal point:

$$\mathbf{u} = (\min E_p, \min E_q) \tag{15.6}$$

  where the minima are computed over the entire domain of the loss functions E_p and E_q. Then the combined loss function is defined as the weighted Chebyshev distance to the ideal point $\mathbf{u}$:

$$S_{WCS} = \max\left\{\lambda_1 \left(E_p - u_1\right), \lambda_2 \left(E_q - u_2\right)\right\} \tag{15.7}$$

  where again $\lambda_1, \lambda_2 \geq 0$. In our case, the ideal point is taken to be the null vector $\mathbf{u} = (0, 0)$, since the loss functions E_p and E_q are non negative.

Once the selected combined loss function S has been calculated, the gradient of the combined loss function with respect to a neural weight must be computed, $\frac{\partial S}{\partial w}$, in order to apply stochastic gradient descent for the update of the neural weight. Stochastic gradient descent variants and optimizations can be applied at this point.

15.3 Experiments

The content of this section reports the experiments that were carried out. The proposed methodology is applicable to any kind of regression network whose optimization layer is based on the minimization of a cost function that compares the input and the output of the network. Thus, two sets of experiments with two different convolutional neural networks were carried out, whose description is made in Subsections 15.3.1 and 15.3.2, including the description of the used datasets as well as the low-resolution image generation procedure. In addition to this, in Subsection 15.3.3 the third experiment for anisotropic generated data is presented in order to study the performance of the proposed methodology. The metrics employed to evaluate the performance of the proposal are detailed in Subsections 15.3.4 and the parameter tuning of the proposals is performed in Subsection 15.3.5. Subsection 15.3.6 details the statistic analysis we carried out. Finally, Subsection 15.3.7 sums up the outcomes of the experimental analysis.

Four different optimization models have been tested in each experiment: the standard squared Euclidean norm ($p = 2$), which is the control algorithm, the best p-norm found in the parameter selection (Subsection 15.3.5), as well as the two proposed multiobjective scalarization methods, WSS and WCS. All the experiments were carried out on a 64-bit Personal Computer with a six-core Intel i7 3.50GHz CPU, 64 GB RAM, with a GPU Nvidia GTX Titan, with 12GB of dedicated memory. With the exception of the neural network execution, the low-resolution image generation and performance analysis were run on MATLAB R2019a, using default parameters.

15.3.1 Experiment 1: SRCNN3D

Firstly, we make use of the SRCNN3D deep neural network (Pham et al., 2017), which is a convolutional neural network that carries out the super-resolution of three-dimensional MR images.

SRCNN3D is based on the application of three blocks of convolutional layers successively, comprising Rectified Linear Unit (ReLU) layers. The method first creates a pre-interpolated image $I(\mathbf{X})$, where $\mathbf{X}$ is the input LR image. Then, the net computes a super-resolved HR image $\hat{\mathbf{Y}}$ by the minimization of the squared Euclidean loss between the output of the CNN and the original HR image $\mathbf{Y}$.

$$f_{SqrEuc} = arg\,min_F \sum ||\mathbf{Y} - \hat{\mathbf{Y}}||^2 \tag{15.8}$$

where $F(I(\mathbf{X})) = \hat{\mathbf{Y}}$.

This network is trained using overlapping patches extracted from a set of HR reference images. A down-sampling and up-sampling are applied to each patch and a set of pairs input-target is created to learn an end-to-

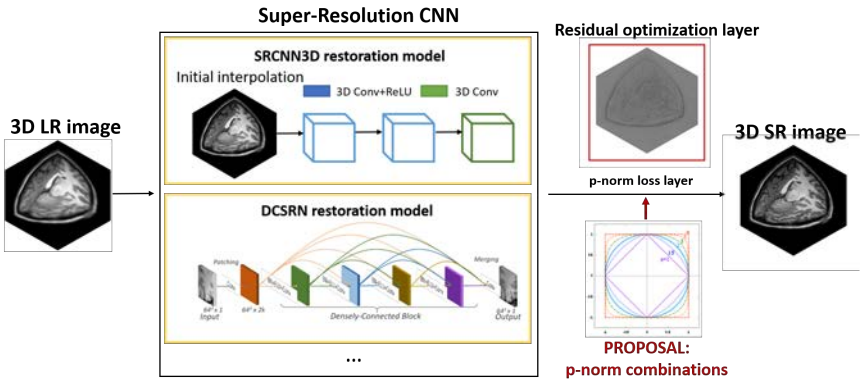


Figure 15.1: Scheme of the proposed model: the LR image is fed into a convolutional neural network with a modified loss layer, producing an optimized SR image. These networks are based on the minimization of the residue between the original HR image and the output of the network.

end function between low and high-resolution images. Specific details of the implementation of this network can be found in the literature (Pham et al., 2017).

A scheme of the operation of the network is shown in Figure 15.1. Our proposal[1] consists of the substitution of the squared Euclidean cost function f_{SqrEuc} by a cost function based on the ℓ_p-norm as described in Section 15.2.

15.3.1.1 OASIS dataset

In order to carry out an adequate analysis of the performance of each model, it was necessary to provide a large dataset of MR images. Nowadays, the number of public datasets has increased, although it is still hard to find the ideal images to be processed since they usually have images of both control and pathological subjects. For this experiment, we considered the OASIS-1 dataset, consisting of a cross-sectional MRI Data of 416 subjects aged 18 to 96. Data were acquired on a 1.5-T Vision scanner with a $1.0 \times 1.0 \times 1.25\, mm^3$ voxel resolution over a FOV of $256 \times 256\, mm$.

A total of 220 T1-weighted MR images of the dataset were considered for the evaluation of the proposed models, which correspond to indices from 0001 until 0240 of type MR1 (patient's first visit), except the image 0080.

Since the whole dataset contains high-resolution images only, low-resolution images were created from the high-resolution ones and fed into the networks. As is stated in (Greenspan, 2008), the observation model is usually decomposed into a linear downsampling operator after a space-invariant blurring

[1]The source code and demo of the proposed approach will be published in case of acceptance.

model as a Gaussian kernel with the full-width-at-half-maximum (FWHM) equal to slice thickness. SRCNN3D is based on this model. For that purpose, the following procedure is applied. Firstly the HR images were adequately cropped to make the image dimensions divisible by the zoom factor. Then, a 3D Gaussian filter with a standard deviation equal to 1 is applied. Finally, *imresize3* MATLAB function was used to perform a 3D cubic interpolation to obtain the LR image. This is a standard procedure to generate the LR versions of HR images for the evaluation of MRI super-resolution algorithms, as seen in (Shi et al., 2015; Du and He, 2019; Zeng et al., 2018).

15.3.1.2 Training procedure

The SRCNN3D has been developed using the Caffe package (Jia et al., 2014) on a Python framework. In this work, a training over 50000 iterations was carried out for each model as well as for the parameter selection. We considered this value to cover a large enough number of epochs that allows the network to converge properly and without taking too much time because each training takes around 12-14 hours to complete. The rest of the network hyper-parameters were set to default: momentum of 0.9, learning rate of 0.0001 and batch size of 256, using Stochastic Gradient Descent (SGD) for model optimization. In order to make the experiments replicable, we set the pseudorandom seed in the Caffe engine to the value 1701.

As described in Subsection 15.3.1.1, from a total of 220 images, the first 120 were used for training. Specifically, the first 100 images were used to train the network and the next 20 ones were used as a validation set to monitor the error curves. The remaining 100 images were used for testing. We divided each training, validation and testing sets into 10 folds with an equal number of images into them (10, 2 and 10 images, respectively) in order to carry out the statistical analysis described in Section 15.3.6. Although it may seem that the proportion training/testing is inadequate, the SRCNN3D model is patch-wise based and extracts around 15000 samples from each training image in order to have a sufficient number of inputs, while the testing is carried on the whole image without patch extraction.

Furthermore, taking advantage of that the SRCNN3D network can be trained for multiple scale factors at the same time, zoom factors 2, 3 and 4 were employed in our analysis. For it, the triple amount of patches was extracted from the training dataset and they were given to the network.

15.3.2 Experiment 2: DCSRN

In the second set of experiments, we make use of the 3D Densely Connected Super-Resolution Network (DCSRN) (Chen et al., 2018c), which is focused on the super-resolution of three-dimensional MR images.

DCSRN is based on a densely-connected block. The network starts with

a convolutional layer applied to the input image, and the output is fed to densely-connected block with 4 units, composed by a batch normalization layer and an exponential linear unit activation followed by a convolutional layer. In the end, a convolution is applied before providing the final SR image.

This network is patch-based so it is faster in training, back-propagation is more efficient, and the model is smaller. The patch size provided to the network is $64 \times 64 \times 64$. Another advantage is that there is less over-fitting during training. Specific details of the implementation of this network can be found in the literature. A scheme of the operation of the network is shown in Figure 15.1.

15.3.2.1 HCP dataset

In this set of experiments, the Human Connectome Project (HCP) (Van Essen et al., 2013; Marcus et al., 2011) was employed, consisting of a great amount of neuroimaging data ranging from structural MRI, functional MRI and diffusion tensor imaging (DTI), from multiple sites. Concretely, we used the HCP Young Adult 1200 Subjects Data Release, which includes 1113 structural MR scans acquired on a 3-T Siemens scanner. The image size is $256 \times 320 \times 320$ with $0.7 \times 0.7 \times 0.7\, mm^3$ voxel resolution. The first 600 T1-weighted MR raw images (no preprocessed) of the dataset were considered for the evaluation of the proposed models.

The low-resolution images were created in a different manner from the one used for SRCNN3D. A Fourier-based procedure is applied (Paschal et al., 2017): first, the Fast Fourier Transform (FFT) is computed on the HR images, then the resolution is degraded by zeroing the outer part of the 3D K-space, and finally, the inverse FFT is applied. The resulting LR image has the same size as the HR one but avoiding the generation of artifacts and it follows the real MR acquisition process. The raw images from HCP were treated as the ground-truth and we lowered the spatial resolution by a factor of 2 in each phase encoding direction to obtain the LR images, that are degraded like using $zoom = 4$.

15.3.2.2 Training procedure

The DCSRN has been developed using Tensorflow 1.8 on a Python 3.6 framework. The training was carried out with the default parameters set by their authors, over a total of 49000 iterations and a batch size of 2. This number of evaluations was empirically deduced monitoring the loss curves until they do not change. Adam optimizer with a learning rate of 10^{-5} was used for model optimization.

From the 600 images of HCP1113 dataset, the first 500 were used for training and the remaining 100 images were used for testing. As well as

described in Subsection 15.3.1.2, data were divided into training and testing sets into 10 folds with 50 and 10 images, respectively to carry out the statistical analysis described in 15.3.6. The DCSRN model is patch-wise based and 200 patches of $64 \times 64 \times 64$ voxels are randomly extracted from each training image in order to have a sufficient number of inputs. For testing, the image was first padded with zeros and then split into cubes of the mentioned size in a 3D sliding window manner using a stride of size 32. Therefore, the reconstructed patches obtained by the networks are merged by averaging the overlapped cubes.

15.3.3 Handling anisotropic data

There are many cases where isotropic low resolution structural brain MRI is uncommon. When such MR images are acquired, for example, a T2 or FLAIR modality, they typically retain a high in-plane resolution to provide sufficient quality data for radiologists to interpret across slices. Thus, the resolution is sacrificed in the through-plane direction and the voxel sizes become anisotropic.

In order to handle this type of image, the SRCNN3D network was used because it has the ability to restore an image in one plane without carrying out an extra training. A set of images of different databases were selected and low-resolution images were generated artificially by extending the voxel size in the last plane by a factor of 2, 3 and 4. Thus, if the image has voxel resolution $1 \times 1 \times 1mm^3$, for each image three LR versions were created of voxel size $1 \times 1 \times 2mm^3$, $1 \times 1 \times 3mm^3$ and $1 \times 1 \times 4mm^3$.

The images used in this experiment are four:

- Images 10 and 11 (named as MPRAGE10 and MPRAGE11 resp.) of the Kirby 21 dataset (Landman et al., 2011). These data were acquired using a 3-T MR scanner with a $1.0 \times 1.0 \times 1.2\,mm^3$ voxel resolution and size $170 \times 256 \times 256$.

- An image of the Medical Research Center of the University of Málaga (CIMES) acquired with size $256 \times 256 \times 190$ and $0.93 \times 0.93 \times 1.0mm^3$ voxel resolution (named as CIMES).

- An image of the IBSR dataset (Worth, 2010) with image size $256 \times 256 \times 128$, with $1.5 \times 1.0 \times 1.0mm^3$ voxel resolution (named as IBSR).

.

15.3.4 Performance measures

In order to evaluate the performance of the proposed model three different quality measures were employed: Peak Signal-to-Noise Ratio (PSNR),

Structural Similarity Index (SSIM) (Wang et al., 2004) and Bhattacharyya coefficient (BC) (Bhattacharyya, 1946).

First of all, PSNR focuses on the intensity values obtained from the algorithm when it is compared with the ground truth image. The unit of measurement is dB (decibel), where higher is better. It is defined as follows:

$$PSNR = 10 \log_{10} \left(\frac{peak^2}{||\mathbf{Y} - \hat{\mathbf{Y}}||^2} \right) \tag{15.9}$$

where *peak* is the maximum possible value of the image and $\mathbf{Y}$, $\hat{\mathbf{Y}}$ are the GT image and the predicted SR image, respectively. The PSNR accumulates the voxel errors provoked by incorrect estimation of the SR image.

On the other hand, SSIM focuses on structural similarities between images, returning a value between 0 and 1 (higher is better). This measure permits to check whether the edges are correctly preserved and it is formulated as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{15.10}$$

where $\mu_x$and μ_y are the mean value of images x and y, σ_x and σ_y are the standard deviations of images x and y, σ_{xy} is the covariance of x and y, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ (default values were used: $L = 1$ is the dynamic range, $k_1 = 0.01$ and $k_2 = 0.03$).

Finally, the BC measures the closeness of the two discrete pixel probability distributions P and $\hat{P}$ corresponding to the Ground Truth (GT) and restored images with values in the range $[0, 255]$:

$$BC = \sum_{j=0}^{255} P(j)\hat{P}(j) \tag{15.11}$$

where $BC \in [0, 1]$ and higher is better.

From a qualitative point of view, it is useful to analyze the residual images obtained by the subtraction of the GT image Y and the super-resolved one $\hat{Y}$:

$$ResI = |Y - \hat{Y}| \tag{15.12}$$

The best performance is such that the residual image is the zero matrix. The constant 0.5 was added to the residual images for the sake of clarity, thereby obtaining gray images.

15.3.5 Parameter selection

Our main aim was to attain the best generality at the time of tuning up the parameters of the proposals, so a set of images different from the ones used for training and testing were used. In the case of the SRCNN3D network

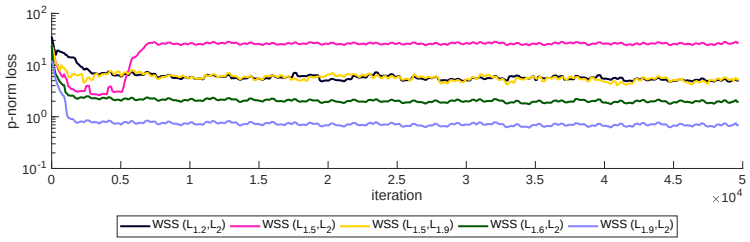


Figure 15.2: Training loss curves of the WSS model optimization (logarithmic scale) using SRCNN3D network.

three different images from 3 different datasets were used to fine-tune the model parameters of each cost function:

- Image 0080 of the OASIS-1 dataset.
- Image 01 of the Kirby 21 dataset (Landman et al., 2011).
- A normal brain T1 image of Brainweb[2] simulated database, acquired with slice thickness $1\,mm$, $0\,\%$ of noise level and $RF = 0$.

Regarding the DCSRN network, the MGH HCP Adult Diffusion dataset was used, which comprises 35 young adult structural scans using the MGH Siemens 3T Connectome scanner.

PSNR, SSIM and BC measures of the above-presented images were computed and a ranking was established sorting each tested parameter according to its performance with respect to each image. The assigned points were accumulated among all images and the lowest scores mean that the network is better. Thus, there are two kinds of parameters to be tuned: the ℓ_p-norms and the weights.

Firstly, we performed a set of experiments fixing weights and varying the ℓ_p-norms. For the SRCNN3D, there were many cases where the convergence of the network was not achieved and the best performances were always obtained with $p = 2.0$ and $q = 1.9$ norms, which combines respectively the stability of the gold standard cost function and the best ℓ_p-norm obtained in our previous work (Thurnhofer-Hemsi et al., 2019). An example of training loss curves for the WSS model is shown in Figure 15.2. In the case of DCSRN, we found a better performance of the $p = 2.0$ and $q = 1.7$ norms.

Secondly, for both WSS and WCS cost functions a set of weight values λ_i were tested in order to find the best configuration. Both WSS and WCS methodologies depends on two variable weights λ_1, λ_2. As the training takes a long time to be completed we aimed to simplify the parameter optimization by making some assumptions, which are detailed next. In the case of

[2] http://mouldy.bic.mni.mcgill.ca/brainweb/

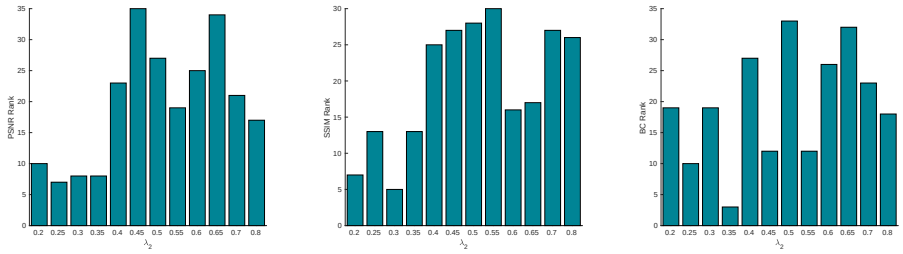


Figure 15.3: WSS model optimization for SRCNN3D: PSNR, SSIM, and BC rankings across the three tuning images are shown varying λ_2, with $\lambda_1 = 1 - \lambda_2$.

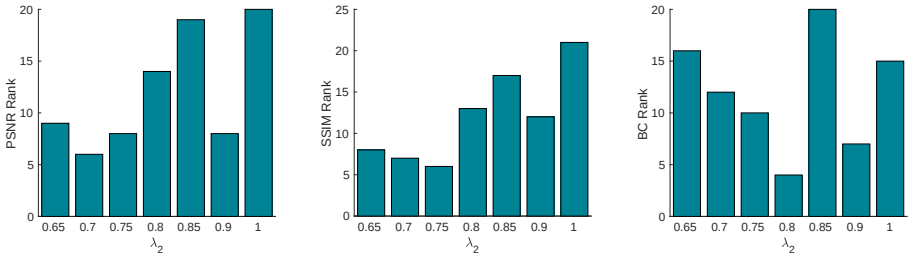


Figure 15.4: WCS model optimization for SRCNN3D: PSNR, SSIM, and BC rankings across the three tuning images are shown varying λ_2, with $\lambda_1 = 1$.

the weighted sum scalarization method (WSS), we are dealing with a linear combination of two ℓ_p-norms, so we assumed that $\lambda_1 = 1 - \lambda_2$, thereby removing one variable. In Figure 15.3 we can find the tested values for λ_2, from 0.2 until 0.8. We found that the best PSNR and SSIM ranks are achieved for middle-low values of λ_2 (0.25-0.35), although values around 0.5 are also fine for BC measure.

The Weighted Chebyshev Scalarization (WCS) is based on the maximum value between the two ℓ_p-norms. Thus, a similar scale between the ℓ_p-norms cost functions is essential to avoid the scalarization to be dominated always only by one of the cost functions. Assuming that $\lambda_1 = 1$, we only need to fix λ_2 to balance both norms. Figure 15.4 collects the best range of values, showing us that the best ranks are reached between 0.7 and 0.8 depending on the considered quality measure.

In order to make a reasonable selection of the best λ_2, the mean and standard deviation among the three measure ranks were computed, as shown in Figure 15.5. That is, the rank values depicted in Figure 15.3 (WSS model) are averaged for each value of the parameter, and the same was done with the results of Figure 15.4 (WCS model). The best parameter is marked in magenta, considering that the lowest mean rank is the best option to achieve good performance for all the quality measures.

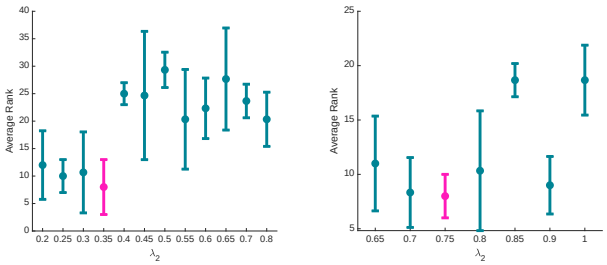


Figure 15.5: Mean and standard deviation of the rank values computed among the PSNR, SSIM and BC ranks for both models: WSS, with $\lambda_1 = 1 - \lambda_2$ (left) and WCS, with $\lambda_1 = 1$ (right).

Table 15.1: Parameter selection of the proposed cost functions.

Model	**Parameters**	
	SRCNN3D	DCSRN
WSS	$p = 2.0,\ q = 1.9,\ \lambda_1 = 0.65,\ \lambda_2 = 0.35$	$p = 2.0,\ q = 1.7,\ \lambda_1 = 0.65,\ \lambda_2 = 0.35$
WCS	$p = 2.0,\ q = 1.9,\ \lambda_1 = 1.00,\ \lambda_2 = 0.75$	$p = 2.0,\ q = 1.7,\ \lambda_1 = 1.00,\ \lambda_2 = 0.45$

In Table 15.1 the final configurations of the models based on our previous analysis are summarized. With respect to the DCSRN network, the configurations of the WSS and WCS methods were established taking into account the previous analysis. Thus, the WSS parameters were set up with the best two p-norms found and with the same weights. The WCS optimization also used the $p = 1.7$ norm but modifying the weight λ_2 in order to put the norm on the same scale as the squared Euclidean norm.

15.3.6 Significance analysis

First, a Friedman aligned ranks test (García et al., 2010; Derrac et al., 2011) is performed in order to check whether at least two of the methods represent populations with different median values, i.e. the methods have significantly different performance. This technique is a similar version of the Friedman test that can be used under the same circumstances, although the Friedman aligned ranks test is appropriate where the number of methods to be compared is low.

In this technique, if we have NK test images, we can do K runs with N test images per run, so that the images of each run are different. Then we compute the mean of the performance measure for the N test images of each run. This way we get K mean performance values for each method and performance measure. Then, we calculate the difference between the performance obtained by a method and the mean performance value computed over all methods. This step is repeated for all methods and datasets. The

resulting differences are then ranked from 1 to 4 (since there are four methods), where the best performing method is assigned 1 point and the worst performing method is assigned 4 points. After that, the K ranks associated with each method are accumulated, so that each method obtains an accumulated rank between K and $4K$. Finally, the test statistic is computed:

$$z = \frac{(R_i - R_j)}{\sqrt{k(n+1)/6}} \tag{15.13}$$

where R_i, R_j are the average rankings of the compared methods.

Then, if the obtained p-value is smaller than the level of significance $\alpha = 0.05$, then the null hypothesis is rejected, i.e. there are at least two methods with significantly different performance. In our work, $N = 10$ and $K = 10$. We used the original software published by the authors[3].

Additionally, box plots were generated, one box plot per performance measure, where each method is associated to a box, and the N mean values of the performance measure for a particular method are used to generate the box, i.e. each of the boxes of the box plot is drawn from N samples.

15.3.7 Results

First, we evaluate the proposed methodology from a quantitative point of view for each experiment. For each quality measure and each zoom factor, a Friedman aligned ranks test was carried out to measure between the different cost functions tested. A total of 10 mean values corresponding to the 10 test repetitions were computed for each method and passed to the Friedman aligned ranks test. The methods were ranked assigning 1,2,3 or 4 points for each repetition and the accumulated results are the ones presented in the following tables.

15.3.7.1 Results of Experiment 1

PSNR aligned Friedman ranks are shown in Table 15.2. The lower values for all zoom factors, i.e the best, are achieved always by the model based on a unique ℓ_p-norm with $p = 1.9$. Moreover, the p-value is lower than 0.05 for all cases, which means that the methods are not the same.

There is a bit of variety for SSIM and BC measures, whose statistical analysis is summarized in Tables 15.3 and 15.4. For larger scale factors (3 and 4), where the network needs to be more precise to recover the voxel's information, the weighted Chebyshev scalarization is clearly the best cost function, achieving the lowest ranks of all the Friedman analysis performed. This means that the WCS is more suitable for recovering the structural features of the MR image than both the usual squared Euclidean norm and

[3] https://sci2s.ugr.es/sicidm

Table 15.2: Friedman Aligned Rankings of the methods for PSNR measure and for zoom factors 2, 3 and 4, computed for the SRCNN3D network. The last row shows the probability value to reject the null hypothesis.

PSNR	**Ranking**		
SRCNN3D	$zoom = 2$	$zoom = 3$	$zoom = 4$
WCS	23.4000	20.0999	20.5999
WSS	20.1000	23.3000	22.3000
$p = 1.9$	14.9000	17.5000	14.5999
$p = 2$	23.5999	21.0999	24.5000
P-value	0.0429	0.0414	0.0422

Table 15.3: Friedman Aligned Rankings of the methods for SSIM measure and for zoom factors 2, 3 and 4, computed for the SRCNN3D network. The last row shows the probability value to reject the null hypothesis.

SSIM	**Ranking**		
SRCNN3D	$zoom = 2$	$zoom = 3$	$zoom = 4$
WCS	16.9999	13.1000	11.5000
WSS	23.2000	23.3000	23.7000
$p = 1.9$	14.9000	20.1000	25.6000
$p = 2$	26.9000	25.5000	21.2000
P-value	0.0065	0.0427	0.0440

the $\ell_{1.9}$-norm. As shown in Table 15.3, for scale factor 2 WCS is the second best method and the p-value is significantly lower than 0.05, only surpassed by $p = 1.9$, making this optimization model effective for any case.

The differences in terms of BC are closer. In Table 15.4 all the average rankings have values around 20, although again for zoom factors 3 and 4 the WCS method has the best outcome, corroborated by the p-value. However, we can see that the squared Euclidean norm is the second best cost function. As the BC measures the differences in pixel probability distributions, we can infer that the image histograms obtained by all the tested methods are very similar, thereby avoiding the inclusion of artifacts or anomalous intensity values.

The variances between the 10 runs of 10 different images we executed are depicted as box plots in Figure 15.6, for scale factors 2, 3 and 4. Analyzing first the zoom 2 (first row), in terms of PSNR and SSIM the ℓ_p-norm with $p = 1.9$ the performance is remarkable. The mean and median values (represented as a circle and a line, resp.) overcome the other three cost functions and the variance of the results is quite reduced. Thus, the image intensity values are always very close to the original HR image. There are no meaningful differences between the other methods, although the BC box

Table 15.4: Friedman Aligned Rankings of the methods for BC measure and for zoom factors 2, 3 and 4, computed for the SRCNN3D network. The last row shows the probability value to reject the null hypothesis.

BC	**Ranking**		
SRCNN3D	$zoom = 2$	$zoom = 3$	$zoom = 4$
WCS	20.4000	19.8000	19.5999
WSS	20.3000	20.6000	21.1000
$p = 1.9$	21.1000	21.4000	21.7000
$p = 2$	20.2000	20.2000	19.5999
P-value	0.0412	0.0415	0.0418

plots show higher differences between the mean and the median except for $p = 1.9$. Again, this may indicate that there is an anomalous test image that is better super-resolved on average by $p = 1.9$. However, in median, the best method appears to be WCS. The case of scale factor 3 (second row) is quite similar to the previous one. However, the performance of WCS has improved a bit and the one of WSS has worsened. Looking at the median values, $p = 1.9$ and WCS achieved the best measures over the 10 runs for PSNR and SSIM. When the zoom demand is set to 4 (third row), which is very high, the variance between runs and between methods is reduced. However, the WSC method takes advantage of both the $L_{1.9}$ and L_2-norms to improve the quality of the image in terms of either PSNR, SSIM, and BC. The latest measure has reduced considerably its interquartile range compared to the other methods.

15.3.7.2 Results of Experiment 2

Figure 15.7 summarizes the outcomes obtained by the DCSRN network after modifying its cost function. As explained in Subsection 15.3.2.1, this network was trained by simulating degradation by Fourier transforms similar to zoom factor 4. The most stable method with less dispersion in its results is WSS, which yields better SSIM and BC measures than the squared Euclidean norm. The mean value (circle) of PSNR is on par with the ℓ_2-norm, although in general, the results are not as good as expected. Nevertheless, the results generated by WCS model are higher for PSNR and also quite acceptable for the rest of the measures. The $\ell_{1.7}$-norm has the lowest dispersion but the results are not the best.

The statistical analysis carried out to check if the methods are significantly different is presented in Table 15.5. The average rankings computed by the Aligned Friedman test showed that WCS is the first method for PSNR followed by WSS model, and this one is the best for SSIM and BC. It should be remarked the difference obtained with respect to the ℓ_2-norm, which is

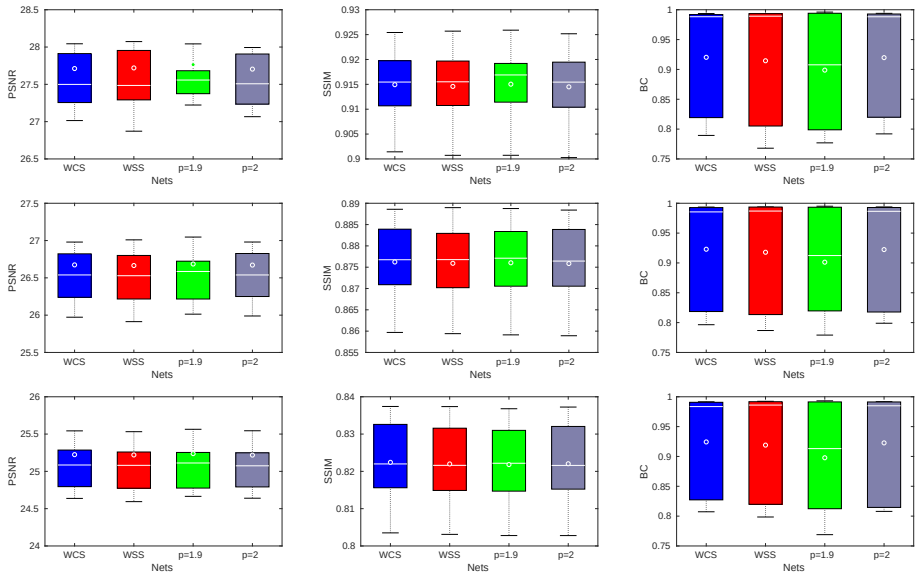


Figure 15.6: Comparison of the PSNR, SSIM and BC for the four models using SRCNN3D network and $zoom = 2$, $zoom = 3$ and $zoom = 4$ (from top row to bottom). Box plots of 10 runs are displayed, where the medians are plotted as horizontal gray lines, while the means are plotted as gray circles.

the last method in the tests. The p-value lover than 0.5 corroborates that the methods are significantly different.

The WSS and WCS results of this set of experiments confirm the idea of improving the performance by the combination of two p-norms. Both cost functions are appropriate for the degradation model based on Fourier transform, which simulates the acquisition of MR images with lower resolution.

15.3.7.3 Anisotropic Super-Resolution

The last set of experiments deals with anisotropic images. The quantitative outcomes of the zoomed images are collected in Table 15.6. The values of the scale factors are referred to the SR applied to the third dimension only. The rows show the results obtained for each of the four tested images and the columns represent the measures obtained for each scale. In blue are highlighted the best values for each measure and zoom factor.

The best optimization method is shared by the $\ell_{1.9}$-norm and WSS. From one side, WSS performs better with MPRAGE10 and IBSR images improving the PSNR and SSIM in some tenths. MPRAGE11 and CIMES have been restored better by the single ℓ_p-norm, although the second-best method is again WSS. This fact indicates that the weighted linear combination of ℓ_p-norms can be a great option to also improve anisotropic images. Regarding

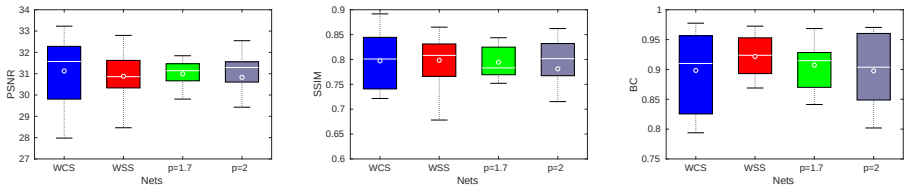


Figure 15.7: Comparison of the PSNR, SSIM, and BC for the four models using DCSRN network. Box plots of 10 runs are displayed, where the medians are plotted as horizontal gray lines, while the means are plotted as gray circles.

Table 15.5: Friedman Aligned Rankings of the methods for PSNR, SSIM, and BC measures computed for the DCSRN network. Last row shows the probability value to reject the null hypothesis.

Model	**Ranking**		
DCSRN	**PSNR**	**SSIM**	**BC**
WCS	18.9999	20.5000	22.2000
WSS	20.2000	19.0000	16.4000
$p = 1.7$	21.5000	20.0999	20.4000
$p = 2$	21.2999	22.4000	23.0000
P-value	0.0393	0.0396	0.0394

the BC measure, either $p = 1.9$, $p = 2$ or WCS reach good values.

On the other hand, the differences between method performances increase when the scale factor applied in the SR process is higher, due to the necessity of recovering more information from the one present in the image. This occurs for every method if we compare them with the squared Euclidean norm, obtaining around 1% of improvement in the quality of the image. We need to remark that the network was only trained for isotropic tasks, so it is easy to think that appropriate training focused only on this type of image may improve substantially the outcomes.

15.3.7.4 Qualitative performance

In terms of qualitative outcomes, three different images of the OASIS dataset are presented. Figure 15.8 shows a three-dimensional perspective with one slice of the sagittal, coronal and axial planes of image numbered as 0174, using an augmentation factor of 2. The differences can be seen in the central part of the image, where the intensity of gray varies from one method to another. If we focus on the residual images, the whitest image should be the best approximation to the original HR image. Here, the method based on the $\ell_{1.9}$-norm clearly outperforms the other methods since there are fewer dark curves in the central part of the image. This means that fewer structures

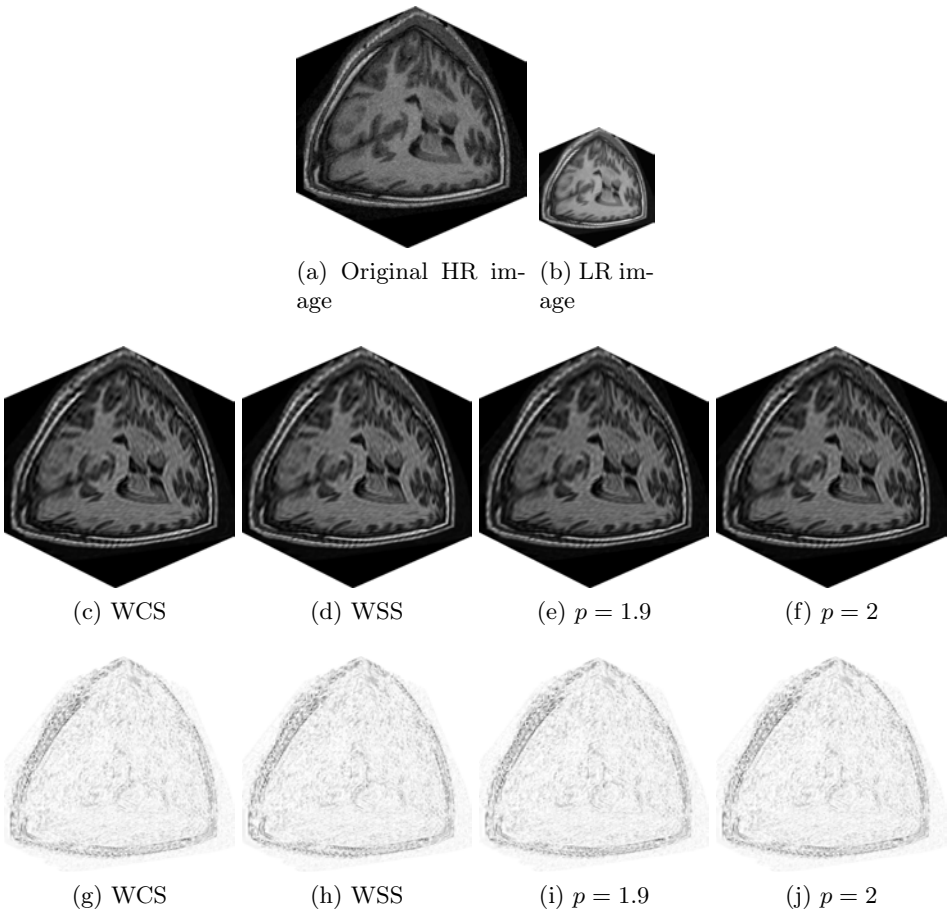
Table 15.6: PSNR, SSIM and BC measures computed for the SRCNN3D network varying the scale factor of the third dimension of the anisotropic LR image. The blue color represents the best method.

Image	Zoom	$zoom = 2$			$zoom = 3$			$zoom = 4$		
	Model	PSNR	SSIM	BC	PSNR	SSIM	BC	PSNR	SSIM	BC
MPRAGE10	WCS	33.4432	0.9580	0.9957	31.9899	0.9426	0.9955	30.8662	0.9266	0.9954
	WSS	33.4923	0.9606	0.9742	32.0180	0.9452	0.9749	30.8850	0.9292	0.9753
	$p = 1.9$	33.4870	0.9605	0.9966	31.9956	0.9448	0.9964	30.8595	0.9287	0.9963
	$p = 2$	33.4587	0.9579	0.9960	32.0001	0.9424	0.9958	30.8737	0.9263	0.9957
MPRAGE11	WCS	35.9565	0.9641	0.9967	34.5699	0.9507	0.9963	33.3593	0.9350	0.9961
	WSS	36.0155	0.9655	0.9579	34.5953	0.9518	0.9583	33.3875	0.9362	0.9579
	$p = 1.9$	36.0548	0.9655	0.9971	34.6155	0.9516	0.9968	33.3990	0.9360	0.9966
	$p = 2$	35.9883	0.9641	0.9968	34.5899	0.9507	0.9964	33.3713	0.9349	0.9962
IBSR	WCS	33.4739	0.9730	0.9269	31.0103	0.9560	0.9254	29.3085	0.9367	0.9268
	WSS	33.5867	0.9733	0.9293	31.0563	0.9561	0.9274	29.3414	0.9368	0.9283
	$p = 1.9$	33.4919	0.9717	0.9273	31.0001	0.9543	0.9255	29.3022	0.9349	0.9272
	$p = 2$	33.5348	0.9731	0.9270	31.0455	0.9561	0.9253	29.3251	0.9368	0.9268
CIMES	WCS	35.5276	0.9568	0.9975	33.4533	0.9306	0.9971	32.0381	0.9026	0.9960
	WSS	35.5835	0.9579	0.9636	33.4834	0.9313	0.9633	32.0693	0.9032	0.9610
	$p = 1.9$	35.6078	0.9581	0.9980	33.4932	0.9315	0.9977	32.0748	0.9036	0.9969
	$p = 2$	35.5522	0.9568	0.9975	33.4691	0.9306	0.9970	32.0455	0.9025	0.9960

were removed. This result matches the previous quantitative analysis, where its values of PSNR and SSIM are the best.

In Figure 15.9 a section of the coronal plane of the OASIS-0177 image is shown. In this case, the restoration factor was 3. Here the effect of the $p = 1.9$ based method is the opposite in terms of similarity structure. The darkest parts of the image are refined while in the original ground-truth image there some gray level irregularities, removing possible abnormalities of the scan or the brain structure. On the other hand, the squared Euclidean norm over-smooths the output of the network with respect to the ideal image. Both WSS and WCS techniques find an equilibrium between both cost functions, as we can see in the middle-left part of the residual images, where the gray is more homogeneous.

Finally, the visual outcome of DCSRN network is analyzed in Figure 15.10. Here a section of the test image with ID 206929 is displayed. The amount of information to be recovered is quite high, provoking distortions in the enhanced image. Focusing on the residual images we can see better the different performance of the optimization methods. WCS method tends to resolve better the intensities because large gray surfaces are clearer than the other methods. On the other hand, fine details are remarked by WSS and $p = 2$ since connections between those homogeneous parts are darker, although they introduce some noise in the image. The $\ell_{1.7}$-norm does not achieve to restore the image adequately. This alternation of performance is present along all the volume, making the quality measures in concordance with the box plot analysis.



(a) Original HR image (b) LR image

(c) WCS (d) WSS (e) $p = 1.9$ (f) $p = 2$

(g) WCS (h) WSS (i) $p = 1.9$ (j) $p = 2$

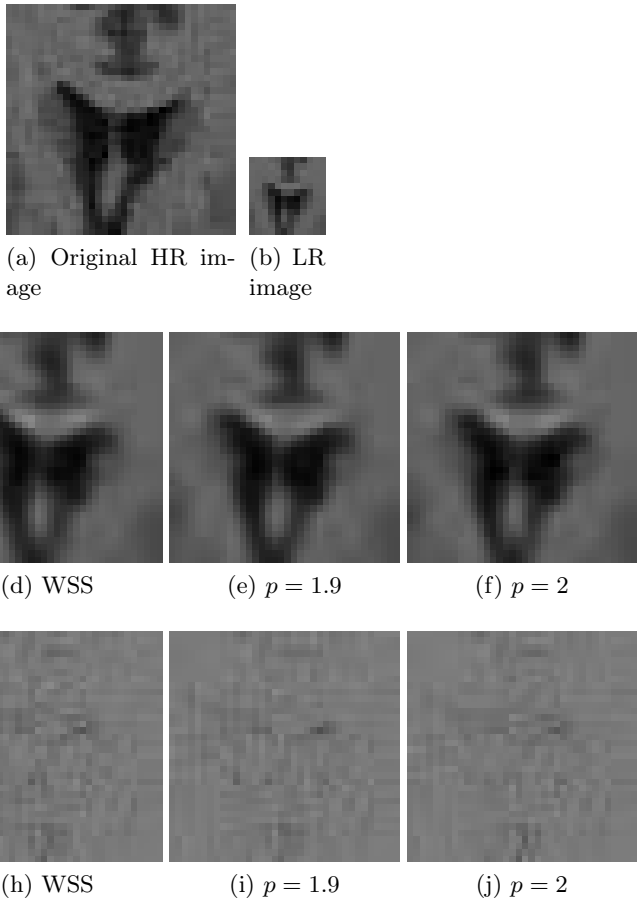
Figure 15.8: Qualitative results for OASIS-0174 image for each model, applied with zoom factor 2. Slices of sagittal, coronal and axial views are showed in a 3D representation. The second row shows the reconstructed image by each algorithm and the third row shows residual images between the reconstructed and the original HR image.

15.4 Discussion

The above-presented experiments have demonstrated that the multiobjective optimization of the cost function makes the SR networks more precise. Two different ways to combine the ℓ_p-norm are proposed, WSS and WCS, although its performance varies depending on the dataset and neural network used.

The WCS performed well with SRCNN3D, the SSIM and BC values improved with respect to the squared Euclidean norm. Nevertheless, the WSS method worked better than WCS with the DCSRN network. In this case, WSS yielded good outcomes for either PSNR, SSIM, and BC, but also

WCS was good restoring the images. There are two factors that may affect the results of the experiments: the type of the neural network and the data used for training.



(a) Original HR image (b) LR image

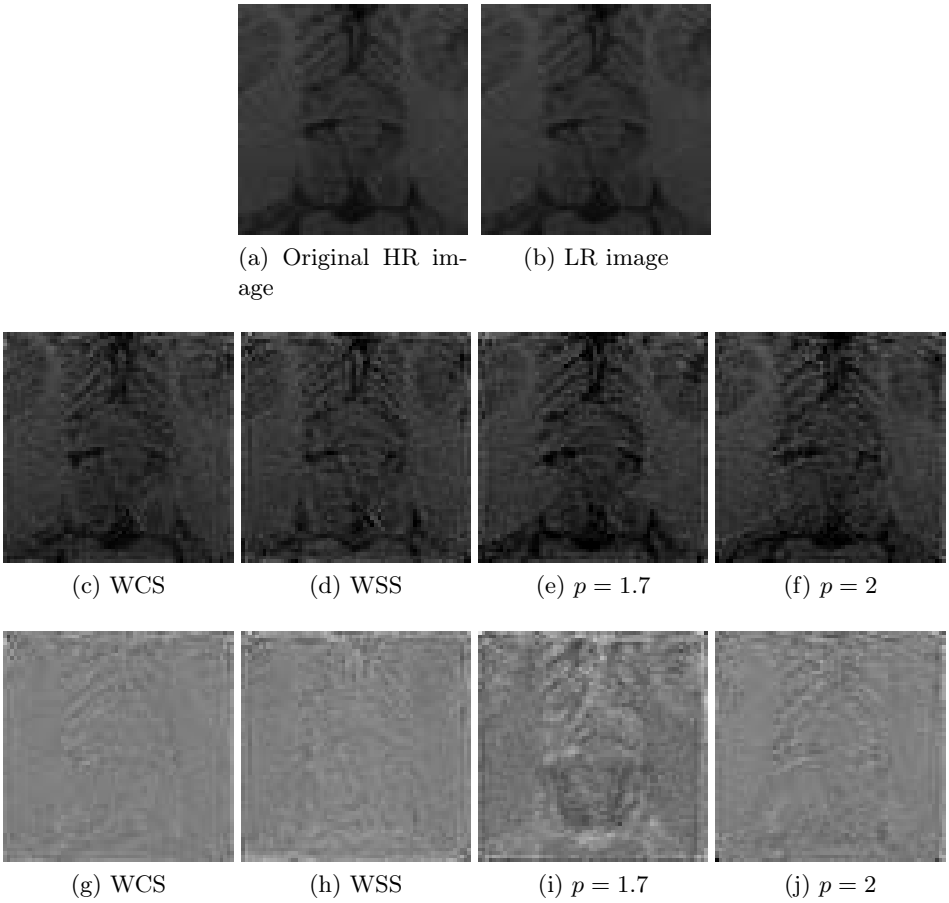
(c) WCS (d) WSS (e) $p = 1.9$ (f) $p = 2$

(g) WCS (h) WSS (i) $p = 1.9$ (j) $p = 2$

Figure 15.9: Qualitative results for a section of the coronal view of the OASIS-0177 image for each model, applied with zoom factor 3. The second row shows the reconstructed image by each algorithm and the third row shows residual images between the reconstructed and the original HR image.

Firstly, the effect of the backpropagation in the learning procedure may be crucial in the performance of the methods. With a small network like SRCNN3D (only three convolutional layers), the Chebyshev scalarization achieved more stability. However, when a larger, densely connected network is trained, this methodology loses efficiency and the weighted sum of norms outperforms the rest of the models. Thus, the different layers of the network learned better the features of the images because they were interconnected. This fact may indicate that the larger the network used, the more efficient might be the combination of p-norms.

Secondly, the amount of data and patch sizes were different in each case. For DCSRN, a larger patch is used, which covers more details of the image, and the minimization of the errors is more effective if the cost function is more complex. The WCS is essentially one ℓ_p-norm that varies depending on the maximum value reached. Thus, when less information is present, that is, smaller patches as the ones used in SRCNN3D, one norm is enough to minimize the error.



(a) Original HR image (b) LR image

(c) WCS (d) WSS (e) $p = 1.7$ (f) $p = 2$

(g) WCS (h) WSS (i) $p = 1.7$ (j) $p = 2$

Figure 15.10: Qualitative results for a section of a reconstructed patch of the image ID 206929 from HCP dataset for each model, applied with zoom factor 4. The second row shows the reconstructed image by each algorithm and the third row shows residual images between the reconstructed and the original HR image.

The qualitative outcomes showed that the scalarization methods can provide refined results. The differences among methods were more noticeable in the DCSRN network than in SRCNN3D. The reason might be the degradation model on which they are based. SRCNN3D carries out an initial

interpolation that can smooth the effect of the restoration, while DCSRN is created to enhance LR images based on a low-pass filtering-like, i.e. the number of voxels are the same and there is not an intermediate blurring effect.

15.5 Conclusions and future works

This work presents a multiobjective optimization model for deep super-resolution neural networks. With the aim of improving the brain magnetic resonance image super-resolution, the usual squared Euclidean loss layer is substituted by combinations of ℓ_p-norm cost functions using the weighted sum and the weighted Chebyshev scalarizations. The optimization function is defined with $p < 2$ to reduce the effect error of extreme values of the errors and enhance the behavior of the training.

SRCNN3D and DCSRN models, and OASIS and HCP datasets were employed for experiments. Three different models were compared to the ℓ_2-norm and results show that the squared Euclidean loss layer is not the best norm in most cases. When we duplicate the resolution, the $\ell_{1.9}$-norm evince good results in all measures and if large scale factors are used, WCS and WSS methods outperform the SSIM and BC values. A Friedman aligned ranks test indicates that the differences with respect to the gold standard are significant at 95%. Qualitatively, image restoration by WCS better preserves structural information.

In future work, we will extend and apply the proposed idea to other machine learning tasks, such as recommendation task and person tracking (Gómez-Silva et al., 2019). The proposed approach could be extended to other neural networks in order to improve the quality of the outputs in any other task like noise removal or segmentation. Moreover, the depth of the network seems to be the key to a correct back-propagation of the proposed cost function errors. An extensive analysis with deeper neural networks may improve the performance with lower values of p. Another line of research is the usage of three or more cost functions to be optimized rather than two. Adding more p-norms might improve the quality of the super-resolved images, but the selection of the p-norms to be included in the set of cost functions to be optimized is a difficult optimization problem in itself. This is why we have restricted our attention in this work to the two cost functions case, while the other cases are left for future extensions of our approach.

Conclusions

Chapter 16

Conclusions and further works

You do not really understand something unless you can explain it to your grandmother.

Albert Einstein

Abstract: This chapter describes a discussion of the previous chapters, and the general conclusions that have been reached after the completion of this Ph.D. thesis, and also indicates a series of possible lines of future research.

16.1 Discussion

The research done in this Ph.D. thesis followed a typical workflow in image pre-processing, which comprises the segmentation of the image, the posterior fitting of some geometrical shape, and the enhancement of the qualitative properties of the image. That said, this thesis has been divided into three main parts, which also coincide to a more significant extent with the different application fields of the proposals.

Part II is focused on the segmentation of the background and foreground of the images. First, Chapters 6 and 7 presented two works that model the panoramic background recorded by a PTZ camera. On the other hand, Chapter 8 described a segmentation system for diabetic skin wounds. The particularity of the works presented in this part is that the ℓ_2-norm and the mean have been used as the principal techniques for implementing the methods.

In Chapter 6, a competitive neural network has been proposed, which is used to provide a solution to the background scene modeling problem. This

neural network is based on the distance error between the pixels and the winning neuron, which is computed as the ℓ_2-norm of their difference. In this case, the Euclidean norm is the most appropriate measure since we are dealing with pixel coordinates in the plane, so the distances are preserved under any rigid transformation. Moreover, the Delaunay triangulation, which is associated with a kind of weighted mean, is used to determine the final color intensity values over each pixel. Since the transition of colors among neighbor pixels is supposed to be smooth, the mean should be a stable measure to model the pixel color distribution. The proposed system yielded good results with several video sequences.

Related to the above system, Chapter 7 describes a neural gas network system for the same goal. The neural gas networks are inspired in a self-organizing map and try to find optimal data representations based on feature vectors. Again, the Euclidean norm is used to preserve the topology of the system, and the barycentric interpolation is the way as the mean is present to generate the colors of each pixel. The particularity of this method is that it requires the ensemble of multiple neural gas networks. Therefore, the computations are more substantial, but the qualitative performance is much better than our previous method.

Chapter 8 focuses on the segmentation of pictures of skin lesions. Although the application is different, the philosophy is similar: we need to distinguish the skin (background) from the wound (foreground). For that purpose, a deep network was used in combination with both pre-processing and post-processing steps. The main difficulty of this segmentation problem was the heterogeneity present among the images, not only in shapes but also in colors, which complicates the functioning of the deep network. Therefore, the pre- and post-processing were crucial to obtaining a good performance. A specific convolution kernel, which is an average over a neighborhood, and the averaged output probabilities of the networks have been the basis of this successful research.

Part III collects three point fitting methods for 2D and 3D geometric shapes. In these cases, more precision was needed, so the ℓ_1-norm and the median have been the metrics employed mostly. The problems of parabola and ellipse fitting were addressed in Chapters 9 and 10, while the evolution to three dimensions was presented in Chapter 11.

First, a method to fit a parabola to a set of bi-dimensional points is described in Chapter 9. Two essential characteristics should be distinguished in this algorithm: the function that measures the error of the points with respect to the parabola, and the cost function to be minimized. The first one is based on the usual Euclidean norm, as they are distances in the plane. The cost function is formulated as a kind of ℓ_1-norm to increase the fitting robustness in the presence of anomalous samples. The optimization by the gradient descent technique yielded better results than the newly published

methods.

Another fitting technique is presented for the case of the ellipse. Unlike the parabola case, there are much more ellipse fitting methods developed by the scientific community due to its interest in many applications, like the ones described in Chapter 2. The creation of a new algorithm becomes a challenge. Therefore, in Chapter 10, we make use of the best ellipse fitting methods to fuse their results properly. This methodology resulted in a robust meta-model very useful when outlying points are present in the training data because although some methods may fail in the fit, some of them generate the adequate ellipse parameters. The trick of our proposal was the use of the ℓ_1 spatial median, which is a median adapted to the Euclidean space, to combine all the methods.

Ending up with this part, the complexity is increased in Chapter 11 offering a framework for ellipsoid fitting. In this case, the proposed idea was very similar to the one exposed in Chapter 9, but adapted to three dimensions. The ℓ_1-norm was used as the cost function with the help of a regularization term to avoid degenerate solutions, and again the useful gradient descent method minimized the function to obtain the ellipsoid parameters. The application of this work on several real-world objects through a stereo 3D camera leads us to the conclusion that the method can be integrated into any low-cost detection system after an adequate calibration.

Part IV is focused on the super-resolution problem, and more specifically, to its application on brain Magnetic Resonance images, which have been described in Chapter 3. That is because super-resolution is a topic addressed by many researchers, but most of them are focused on RGB images. The requirement of high levels of precision in the image enhancement process of medical images is a maxim to avoid bad diagnostics. Chapters 12-15 proposed four SR methods applied to the enhancement of brain images.

Thus, Chapter 12 started providing a SR meta-model based on the combination of multiple shifts of the image. Similarly, like the neural network presented in Chapters 6 and 7, the mean value of the unshifted images demonstrated high efficacy in minimizing the error between the original and the super-resolved images. Therefore, we can conclude that if the data values are similar, then the mean acts as a good estimator of the voxel values distribution generated by the shiftings when the neural network cost function is based on the traditional ℓ_2-norm.

MR images may also be affected by the presence of heavy noise that can affect the SR process. Chapter 13 described an algorithm based on median filters to both remove noise and increase the resolution of the image. The originality of this work resides in modeling the image data by a probabilistic distribution and distribute it among several tilings of the 3D space. Then, they are filtered by the mid-sample median in order to obtain the minimum error. The advantage is that it can be applied to any 3D image or noise type,

although the computational processing time has to be improved.

Finally, this research went into the cost functions of the SR deep networks. The idea proposed in Chapter 14 intends to improve the performance of the network by minimizing a different ℓ_p-norm from the usuals ℓ_1 and ℓ_2. That required a new implementation of the baseline algorithm used in Chapter 12 in order to incorporate the gradients of the ℓ_p-norm. Although differences are small, we found that values of p between 1 and 2 may yield higher performances. The reason to choose this interval is that the values in $0 < p < 1$ do not define a norm in $\mathbb{R}^n$, so continuity and differentiability are compromised. Also, values larger than two would have the opposite effect, enhancing outlying values.

The work presented in Chapter 15 went further and tried combining two ℓ_p-norms with their corresponding weights. Thus, the properties of both ℓ_1 and ℓ_2 norms might enhance the quality of the resulting images. After testing with two different networks and databases, we found out that the combination of norms depends on the architecture of the network more than the dataset employed. The precision of the minimization yielded better quantitatively and qualitatively results.

16.2 Conclusions

This Ph.D. thesis covered several topics of image processing by the use of several computational intelligence techniques. General meta-models, as well as specific applications, have been developed for their application in three different areas: video-surveillance, geometric fitting, and medical imaging. For that purpose, optimization techniques and artificial neural networks based on the ℓ_1, ℓ_2, and more generally, ℓ_p norms have been used to improve the state-of-art methods.

Starting from the use of classical computational intelligence techniques such as gradient descent or simple neural networks (Chapters 6, 7, 9-11, and 13), the emergence of deep learning in the middle of this research has introduced a new challenge to be addressed (Chapters 8, 12, 14, and 15). The studies carried out were varied, but with the same question in mind, how the optimization can be improved. Summing up all the research done, it can be concluded that the optimization process is the key to obtaining one performance or another, depending on the goals of the work. By last, we showed that the proposed techniques could be applied to any field of image processing, which indicates the breadth and applicability to this area of the Science.

The traditional Euclidean norm continues being used due to its simplicity and useful properties. The works presented here showed great efficiency, in special when different information sources need to be fused. That is because of its elementary interpretation of geometrical distance of a vector. When

the distances are reduced, the performance is going to increase.

The robustness against anomalous situations was achieved through the absolute distance, i.e. the ℓ_1-norm. Outlying points or pixel values or more sensitive in the computation of the overall error, making this norm very effective for the reduction of noise.

The benefits of the ℓ_p-norm have to be studied. One can look for an equilibrium between noise reduction and good optimization, although there is not a 'magic' value; it depends on the application. Nevertheless, the weighted combination of several of these norms (or maybe combination of mean and median) may produce better results. Again, the nature of the model to be optimized is tightly related to the selected p values and their weights.

To conclude, the development of these methods has allowed their publication in many prestigious research journals indexed in the first quartile of the *Journal Citation Reports (JCR)* ranking and conferences which appear in the *GII-GRIN-SCIE (GGS) Conference Rating* ranking. The realization of this thesis has also enriched my knowledge of quite branches of computer science, opening my mind to new challenges.

16.3 Future lines of research

The previously commented conclusions lead us to a collection of future lines of research intending to extend these works for more fields and with a major improvement of state of the art. Some of these lines are the following:

- To design new algorithms for segmentation in specific context scenarios, where those are affected by external factors such as 'dynamic' backgrounds or noise. The ℓ_2-norm and the mean seem to be insufficient for harder optimization problems. Thus, other techniques should be studied for their applicability to both video and image segmentation.
- To develop fitting methods for other geometrical shapes of interest, such as paraboloids or any others. Related to this, it is essential to integrate the fitting methods into a detection system so that they might be used for the detection of anomalous objects or pathologies. An automatic system that can detect and classify multiple geometric figures is also interesting.
- To adapt and apply the proposed SR meta-models for any kind of image, both RGB or medical. Since the proposed methods worked well in our studies, they may also enhance the performance of the novel methods in other fields.
- To reduce noise in PET images, which in itself is a noisy technique, and which would either reduce the dose of radioactivity received by

the patient (maintaining image quality) or improve said image quality. Another important application would be dynamic PET, which is an extremely noisy technique that allows us to understand the pharmacokinetic properties of tracers and to estimate the pharmacokinetic parameters themselves. These are usually associated with characteristics of medical or biomedical interest, such as the density of a certain neuroreceptor in the brain or tumor molecules in different organs.

- To improve the state-of-art deep networks by the use of adapted cost functions, either the ℓ_p-norms or any other error measure like SSIM. Also, the inclusion of the median transform filter in a deep network would be fascinating to analyze. In general, the problem of both the Rician noise removal and the resolution enhancement in MR has a long way to go.

- To develop multi-objective optimization methods based on more of two ℓ_p-norms. The results are promising, and adequate analysis of the problem might allow the use of more norms in order to find the best equilibrium and precision in the optimization process.

- To study, from a theoretical point of view, the convergence and efficacy of the optimization methods using the ℓ_p-norm.

- To create a skull stripping method based on the algorithms implemented in this research. This procedure is very demanded, where the background segmentation is needed to fit an ellipse then to separate the skull from the brain. Then image SR can be applied to enhance the brain structures.

Appendices

Appendix A

Publications Supporting this Ph.D. Thesis

Abstract: In this appendix, we summarize the information associated with the publications supporting this Ph.D. thesis. For journals, the *Journal Citation Reports (JCR)* ranking has been used, while for conferences, the *GII-GRIN-SCIE (GGS) Conference Rating* and *CORE* ranking have been employed.

Table A.1 collects the distribution of publications broken down by its type and rating. The conference publications have been classified with both *GGS* and *CORE* rankings, having a total of 4 rated publications within this type. In general, during this Ph.D. thesis, 11 works have been published.

Table A.1: Publications supporting this Ph.D. thesis.

Type of publication	**Rating**	**Publications**	**Total**
JCR Indexed Journals	Q1	4	**6**
	Q2	2	
International Conferences	CORE A	2	**5**
	CORE B	1	
	CORE C	1	
	GGS Class 2	1	
	GGS Class 3	3	
	In Progress	1	

A.1 JCR Indexed Journals

1. López-Rubio et al. (2017): López-Rubio, E., Thurnhofer-Hemsi, K., de Cózar-Macías, Ó. D., Blázquez-Parra, E. B., Muñoz-Pérez, J., and Ladrón-de-Guevara-López, I. (2017). Robust Fitting of Ellipsoids by Separating Interior and Exterior Points During Optimization. *Journal of Mathematical Imaging and Vision*, 58(2):189–210

   - Impact Factor: 1.927 (2017)
   - Category:
     - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE: Q2
     - COMPUTER SCIENCE, SOFTWARE ENGINEERING: Q2
     - MATHEMATICS, APPLIED: Q1
   - DOI: 10.1007/s10851-016-0700-6

2. Thurnhofer-Hemsi et al. (2018a): Thurnhofer-Hemsi, K., López-Rubio, E., Domínguez, E., Luque-Baena, R. M., and Molina-Cabello, M. A. (2018a). Panorama construction for PTZ camera surveillance with the neural gas network. *Expert Systems*, 35(2):e12249

   - Impact Factor: 1.505 (2018)
   - Category:
     - COMPUTER SCIENCE, THEORY & METHODS: Q2
     - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE: Q3
   - DOI: 10.1111/exsy.12249

3. López-Rubio et al. (2018): López-Rubio, E., Thurnhofer-Hemsi, K., Blázquez-Parra, E. B., de Cózar-Macías, Ó. D., and Ladrón-de Guevara-Muñoz, M. C. (2018). A fast robust geometric fitting method for parabolic curves. *Pattern Recognition*, 84:301–316

   - Impact Factor: 5.898 (2018)
   - Category:
     - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE: Q1
     - ENGINEERING, ELECTRICAL & ELECTRONIC: Q1
   - DOI: 10.1016/j.patcog.2018.07.019

4. Thurnhofer-Hemsi et al. (2020a): Thurnhofer-Hemsi, K., López-Rubio, E., Blázquez-Parra, E. B., Ladrón-de Guevara-Muñoz, M. C., and De-Cózar-Macias, Ó. D. (2020a). Ellipse fitting by spatial averaging of random ensembles. *Pattern Recognition*, 106:107406

   - Impact Factor: 7.196 (2019)

- Category:
  - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE: Q1
  - ENGINEERING, ELECTRICAL & ELECTRONIC: Q1
- DOI: 10.1016/j.patcog.2020.107406

5. Thurnhofer-Hemsi et al. (2020b): Thurnhofer-Hemsi, K., López-Rubio, E., Domínguez, E., Luque-Baena, R. M., and Roé-Vellvé, N. (2020b). Deep learning-based super-resolution of 3D magnetic resonance images by regularly spaced shifting. *Neurocomputing*, 398:314–327

   - Impact Factor: 4.438 (2019)
   - Category:
     - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE: Q1
   - DOI: 10.1016/j.neucom.2019.05.107

6. Thurnhofer-Hemsi et al. (2020d): Thurnhofer-Hemsi, K., López-Rubio, E., Roé-Vellvé, N., and Molina-Cabello, M. A. (2020d). Multiobjective optimization of deep neural networks with combinations of Lp-norm cost functions for 3D medical image super-resolution. *Integrated Computer-Aided Engineering*, 27(3):233–251

   - Impact Factor: 4.706 (2019)
   - Category:
     - COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE: Q1
     - COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS: Q1
     - ENGINEERING, MULTIDISCIPLINARY: Q1
   - DOI: 10.3233/ica-200620

A.2 International Conferences

1. Thurnhofer-Hemsi et al. (2017): Thurnhofer-Hemsi, K., Lopez-Rubio, E., Dominguez, E., Luque-Baena, R. M., and Molina-Cabello, M. A. (2017). Panoramic background modeling for PTZ cameras with competitive learning neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2017-May, pages 396–403. IEEE

   - GGS class: 3
   - CORE rating: A
   - DOI: 10.1109/IJCNN.2017.7965881

2. Thurnhofer-Hemsi et al. (2018b): Thurnhofer-Hemsi, K., Lopez-Rubio, E., Roe-Vellve, N., Dominguez, E., and Molina-Cabello, M. A. (2018b). Super-resolution of 3D Magnetic Resonance Images by Random Shifting and Convolutional Neural Networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2018-July

    - GGS class: 3
    - CORE rating: A
    - DOI: 10.1109/IJCNN.2018.8489604

3. Cui et al. (2019): Cui, C., Thurnhofer-Hemsi, K., Soroushmehr, R., Mishra, A., Gryak, J., Dominguez, E., Najarian, K., and Lopez-Rubio, E. (2019). Diabetic Wound Segmentation using Convolutional Neural Networks. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 1002–1005. IEEE

    - GGS class: 3
    - CORE rating: C
    - DOI: 10.1109/EMBC.2019.8856665

4. Thurnhofer-Hemsi et al. (2019): Thurnhofer-Hemsi, K., López-Rubio, E., Roé-Vellvé, N., and Molina-Cabello, M. A. (2019). Deep Learning Networks with p-norm Loss Layers for Spatial Resolution Enhancement of 3D Medical Images. In Ferrández Vicente, J. M., Álvarez-Sánchez, J. R., de la Paz López, F., Toledo Moreo, J., and Adeli, H., editors, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11487 LNCS, pages 287–296, Cham. Springer International Publishing

    - GGS class: Work in progress
    - CORE rating: Work in progress
    - DOI: 10.1007/978-3-030-19651-6_28

5. Thurnhofer-Hemsi et al. (2020c): Thurnhofer-Hemsi, K., López-Rubio, E., Roé-Vellvé, N., and Deka, L. (2020c). Super-resolution of 3D MRIs corrupted by heavy noise with the Median Filter Transform. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1–5

    - GGS class: 2
    - CORE rating: B
    - DOI: Accepted for publication

Appendix B

Resumen en Español

Abstract: Esta tesis trata sobre el procesamiento de imágenes mediante técnicas robustas de inteligencia computacional. El uso y análisis de diferentes normas matemáticas ha sido nuestro objetivo.

Tras presentar el estado del arte centrado en las aplicaciones de nuestras propuestas, se resumen los fundamentos de las modalidades de imagen, con especial atención a la resonancia magnética, y las técnicas de aprendizaje utilizadas en esta investigación, basadas principalmente en redes neuronales. Para terminar, se define el marco matemático en el que se basa este trabajo, las normas ℓ_p.

Siguen tres partes diferentes asociadas con las técnicas de procesamiento de imágenes. La primera parte recoge los desarrollos que tratan sobre la segmentación de imágenes. Dos de ellos son aplicaciones para tareas de videovigilancia e intentan modelar el fondo de un escenario utilizando una cámara específica. El otro trabajo se centra en el campo médico, donde se aborda el objetivo de segmentar las heridas diabéticas de un conjunto de datos muy heterogéneo.

La segunda parte se centra en la optimización e implementación de nuevos modelos para el ajuste de curvas y superficies en dos y tres dimensiones, respectivamente. El primer trabajo presenta un algoritmo de ajuste de parábola basado en la medición de las distancias de los puntos interiores y exteriores al foco y la directriz. El segundo trabajo trata el caso de la elipse, y agrupa la información de múltiples métodos de ajuste. Por último, el problema del elipsoide se aborda de manera similar a la parábola.

La tercera parte está dedicada exclusivamente a la super-resolución de imágenes de resonancia magnética. En uno de estos trabajos, se desarrolla un algoritmo basado en la técnica de desplazamientos aleatorios. Además, estudiamos la eliminación de ruido y la mejora de la resolución simultáneamente. Para finalizar, la función de coste de las redes profundas ha sido modificada por diferentes combinaciones de normas para mejorar su precisión.

Finalmente, se presentan y discuten las conclusiones generales de la investigación, así como las posibles líneas de investigación futuras que pueden hacer uso de los resultados obtenidos en esta tesis.

B.1 Introducción

Hoy en día, el procesamiento de imágenes es un campo de investigación activo. El progreso del área se impulsa continuamente con la ayuda de las matemáticas, la computación y un mejor conocimiento de los órganos específicos del cuerpo humano que intervienen en la percepción de las imágenes. Los avances en el procesamiento de imágenes digitales se reflejan en medicina, astronomía, geología, arquitectura o vigilancia. Así como nuestros ojos perciben la imagen a través de la pupila, las cámaras actúan de manera similar al emular este proceso de acuerdo con un modelo conocido como la "cámara estenopeica". El problema es que uno de los principales problemas que afectan la adquisición de imágenes es la presencia de ruido y la falta de resolución suficiente. Hay muchos tipos de ruido. Entre los más conocidos se encuentran el gaussiano y el ruido impulsivo. Por otro lado, los procedimientos de adquisición a menudo no permiten generar imágenes con suficiente resolución, y esto hace que sea necesario aplicar un método de procesamiento de imágenes para aumentar su resolución. Esta técnica se denomina Super-Resolución (SR), y es un campo de investigación amplio y en continua evolución.

En particular, en imagen por Resonancia Magnética (RM), los datos de píxeles se ven afectados por varios factores degradantes, especialmente la resolución espacial finita y el ruido. La ausencia de resolución puede implicar la pérdida de detalles que podrían ser críticos para el diagnóstico médico. Una dificultad adicional en RM proviene de la naturaleza del ruido presente en estas imágenes, que sigue una distribución de Rice. Ser capaz de mejorar tales imágenes proporcionaría un gran avance para la investigación médica y el diagnóstico de enfermedades.

En el campo de la vigilancia, por ejemplo, además de los problemas anteriores, existen otras tareas como la detección de objetos o la clasificación a partir de imágenes y videos, que están bajo investigación debido a su gran potencial de aplicación. Concretamente, modelar el fondo y el primer plano de un escenario es una tarea compleja debido a la presencia de muchos factores externos. Ser capaz de desarrollar algoritmos robustos que contribuyan, aunque en menor medida o como preprocesados, a resolver estos problemas también es parte del presente trabajo.

Por otro lado, el reconocimiento de patrones en datos bidimensionales o tridimensionales es un problema generalizado en muchos campos de la ciencia y la ingeniería, principalmente debido a la presencia de ruido. Lograr métodos de ajuste robustos frente a situaciones muy ruidosas nos proporcionaría un avance significativo en el procesamiento de imágenes en condiciones extremas y permitiría que los algoritmos de posprocesamiento sean mucho más efectivos. Ejemplos típicos relacionados con esto son el ajuste de formas geométricas o la segmentación, que están presentes tanto en el mundo real

como en la medicina.

Por lo tanto, el problema consiste en tratar de desarrollar métodos que nos permitan resolver varios problemas relacionados a la vez, o poder generar metamodelos que sean eficientes y precisos independientemente del tipo de datos utilizados. Partiendo de varias técnicas clásicas, y también utilizando la potencia de las redes neuronales, este trabajo tiene como objetivo aprovechar sus ventajas y resolver problemas en varios campos de conocimiento. La incorporación de algunas de estas últimas técnicas en nuestro trabajo, como el aprendizaje profundo, aumentaría la posibilidad de lograr algoritmos más precisos.

Formulamos la siguiente hipótesis de investigación: *el uso de variantes específicas de las normas ℓ_p puede mejorar el rendimiento de las funciones de minimización, pudiéndose aplicar a diferentes tareas computacionales y aumentar la solidez de las soluciones.* Nuestro objetivo principal es diseñar, implementar y evaluar nuevos modelos y algoritmos robustos que permitan resolver los problemas anteriores mencionados, que se logrará basándonos en la norma matemática. Para ello, es esencial estudiar la base de las funciones de minimización, así como las propiedades de los espacios vectoriales normados en los que vamos a trabajar. Los algoritmos robustos deben cumplir varios requisitos al ofrecer su solución, además de resolver el problema para el que fueron diseñados. La complejidad de estos métodos no debe ser alta para resolver los problemas que requieren ser implantados en un sistema en tiempo real. Sin embargo, en imágenes médicas, lo más importante es la precisión del algoritmo independientemente del tiempo de cálculo, ya que estos métodos están destinados a ser utilizados para el diagnóstico de enfermedades. Por lo tanto, la calidad del método se determinará según el campo de aplicación.

B.2 Contexto y fundamentos teóricos

B.2.1 Estado del arte

En primer lugar, el problema de la detección de objetos en la videovigilancia se analiza desde la perspectiva de uno de los pasos clásicos de preprocesamiento, la determinación del fondo de un escenario. Además, se analizan las técnicas de segmentación de imágenes para el diagnóstico médico. Después, resumimos los últimos trabajos sobre una elipse, una parábola y un ajuste de elipsoide, así como sus aplicaciones. Para finalizar, se explican los fundamentos de la super-resolución, y se enumeran las técnicas clásicas y las más nuevas.

Algunos de los inconvenientes más importantes para la segmentación del fondo de un escenario son los siguientes: iluminación y sombras, fondos dinámicos, objetos estáticos del primer plano, ruido, movimiento de la cá-

mara, etc. Estos factores hacen que sea esencial desarrollar modelos robustos para separar el fondo adecuadamente del primer plano (Jeeva and Sivabalakrishnan, 2015). Muchos investigadores han propuesto metodologías robustas para reconstruir el fondo basado en medidas de distancia (Yang et al., 2016), o el uso de múltiples cámaras (Cermeño et al., 2018). Sin embargo, un contexto determinante que tener en cuenta en el modelado es el tipo de cámara utilizada en cada situación. Los algoritmos tradicionales de sustracción de fondo suponen que las cámaras son estáticas, y esto lleva a una detección falsa cuando la cámara se mueve (Kim et al., 2013). Se ha llevado a cabo una amplia investigación sobre la detección de objetos para cámaras en movimiento. Algunas propuestas se basan en la agrupación de flujo óptico (Varcheie and Bilodeau, 2011), y otros métodos se basan en la estimación de los parámetros de transformación entre escenas consecutivos (López-Rubio and López-Rubio, 2015). Los enfoques presentados en nuestra investigación se basan en el teselado del fondo (Azzari and Bevilacqua, 2006), que consiste en crear una imagen de fondo panorámica y luego usar una técnica de sustracción de fondo para extraer regiones de objetos en movimiento.

El aprendizaje profundo se ha aplicado para resolver tareas de clasificación y segmentación muy complejas (Pereira Dos Santos and Antonelli Ponti, 2019) sin el uso de ningún método de preprocesamiento de imágenes. Hoy en día, con el aumento de los conjuntos de datos disponibles públicamente, las redes profundas han demostrado un excelente rendimiento en el análisis de imágenes médicas (Litjens et al., 2017). En relación a nuestra investigación, se han publicado trabajos recientes para la clasificación de lesiones cutáneas (Nida et al., 2019), aunque todavía hay un margen de mejora. Estos trabajos se basan en un proceso de dos etapas, por lo que pueden segmentar y extraer características con redes profundas y luego realizar la predicción. Muchas técnicas clásicas se han desarrollado con éxito (Pal and Pal, 1993). Por ejemplo, Comaniciu and Meer (1997) presenta una segmentación basada en el color basada en el algoritmo de cambio medio, siendo muy robusto al extraer las características significativas de la imagen. Otra aplicación reseñable es la segmentación cerebral completa (Fischl et al., 2002) o la segmentación atlas (Lötjönen et al., 2010) en resonancia magnética, donde la precisión tiene que ser muy alta. La segmentación basada en ventanas también se usa mucho en este campo debido a la similitud presente en otras partes del cerebro (Coupé et al., 2011). Proporcionar un sistema robusto para segmentar lesiones con redes profundas sigue siendo un desafío debido a la falta de datos suficientes y su heterogeneidad.

En los últimos años, ajustar modelos primitivos, como una curva cuadrática o una superficie, a una nube de puntos dada es una tarea fundamental en el análisis de imágenes digitales, el reconocimiento de patrones visuales, y la visión por computador. El caso 2D implica ajustar una curva cuadrática a un conjunto de puntos de muestra en el plano (Muñoz-Pérez et al., 2014;

Waibel et al., 2015). Además, la elipticidad y circularidad del conjunto de muestras se pueden medir (Misztal and Tabor, 2016). El caso 3D consiste en ajustar una superficie cuadrática a un conjunto de puntos en el espacio (Sahin and Unel, 2008), que tiene aplicaciones para la reconstrucción 3D (Cross and Zisserman, 1998), estimación de la orientación (Lee and Moore, 2004), problema de correspondencia estéreo restringida (Collings et al., 2009) y reconocimiento de objetos (Cao and Shrikhande, 2002). Dependiendo del tipo de curvas o superficies, la clase de función cuadrática que las define y las técnicas de minimización utilizadas para alcanzar la convergencia del ajuste pueden variar. A continuación, se analizan tres figuras geométricas: parábolas, elipses y elipsoides.

La forma de la parábola está muy presente en el campo de la medicina. Para citar un ejemplo, un estudio sobre dientes (Muhamad et al., 2015). En (Janunts et al., 2015) se propuso un enfoque matemático directo para obtener parámetros relevantes de la superficie corneal. Además, las aplicaciones industriales y de telecomunicaciones son numerosas. En (Späth, 2001), se discutió el problema de ajustar un paraboloide girado a datos medidos dados en 3 espacios. Por otro lado, la detección automática de carriles es una parte esencial de los sistemas de asistencia al conductor en vehículos inteligentes (Narote et al., 2018; Niu et al., 2016). Se puede encontrar una aplicación atmosférica relacionada con la propagación del sonido en (Kampanis et al., 2013). Las ecuaciones parabólicas, aproximaciones de la ecuación de Helmholtz en coordenadas cilíndricas, se usan ampliamente en el contexto acústico subacuático, ver (Dougalis et al., 2008). Finalmente, se muestra un ejemplo en el campo arquitectónico en (Samper et al., 2017), que proporciona un método que espera determinar los mejores ajustes para la forma geométrica de un arco de un edificio patrimonial. Según Bookstein (1979), si las coordenadas del conjunto de datos se transforman, la cónica de mejor ajuste resultante debería coincidir con la cónica de mejor ajuste que obtendría el algoritmo si se hubieran proporcionado al algoritmo las muestras no transformadas. El método de Yu et al. (2009) tiene una función de coste con una interpretación geométrica donde los parámetros son intrínsecos a la cónica y son invariantes con respecto a traslaciones y rotaciones.

Como en el caso de la parábola, los métodos de ajuste de elipses se pueden dividir en algebraicos y geométricos. En un ajuste algebraico, la curva está dada por una ecuación implícita restringida de una cónica. Este ajuste tiene ventajas de implementación y costes de cómputo (Muñoz-Pérez et al., 2014), pero también algunos inconvenientes en la precisión, la interpretación física de los parámetros de ajuste, los errores y la sensibilidad a los valores atípicos. Aunque los algoritmos son eficientes, la solución no siempre es una elipse. La distancia geométrica es empleada por muchos investigadores que utilizan una función de parámetros elípticos; en otras palabras, el error de Sampson (Sampson, 1982). Muchos investigadores han intentado obtener

métodos precisos y robustos, en los que se aproxima la distancia geométrica en función de los parámetros de elipse. Yu et al. (2012) han determinado una función objetivo geométrica considerando que la suma de las distancias desde un punto a los focos es constante. Por otro lado, Kanatani and Rangarajan (2011) ha presentado un método algebraico utilizando la ecuación de elipse implícita con una restricción estipulada. Esto da lugar a una solución algebraica precisa mediante un análisis completo de errores. Liang et al. (2019) usó el método de dirección alterna de multiplicadores en una adaptación de ajuste directo de mínimos cuadrados usando la norma ℓ_p con $p < 2$. Los algoritmos genéticos han sido utilizados por De La Fraga et al. (2007) y Ray and Srivastava (2008) para resolver problemas de optimización del ajuste de elipse. Como ejemplo de la variedad de aplicaciones, Mitchell and Van den Berg (2016) utiliza el ajuste de elipse para analizar la difracción de electrones con el fin de obtener diferentes patrones de materiales policristalinos. Y Liao et al. (2017) usa un método basado en técnicas de detección y ajuste de elipse para la segmentación de la imagen celular.

El elipsoide es una de las primitivas geométricas más utilizadas y tiene una importancia primordial en muchos campos. Muchos dispositivos electrónicos comerciales necesitan resolver problemas de ajuste en 3D. Los teléfonos y navegadores GPS (Lou et al., 2015), relojes y periféricos de videojuegos contienen acelerómetros y magnetómetros que se utilizan para estimar la orientación de un cuerpo en el espacio (Fang and Liu, 2015), calcular trayectorias (Krach and Robertson, 2008), analizar el movimiento (Susi et al., 2013), detectar posibles colisiones con obstáculos (Rimon and Boyd, 1997) y otras aplicaciones de medición innovadoras como el escaneo 3D (Grivon et al., 2013). En el campo de la medicina, se han utilizado elipsoides de ajuste para estimar los diámetros de los nódulos pulmonares (Kubota and Okada, 2005) de las tomografías computarizadas y estimar su diámetro. Otros campos de investigación se centran en la detección o clasificación del cuerpo, como la detección de la punta de la nariz por Sarakon et al. (2013). Hay dos categorías principales de métodos de ajuste (Malyugina et al., 2014), utilizando distancias algebraicas (Blane, 2000) o distancias geométricas (Sung Joon Ahn et al., 2002) según sus definiciones del error. Los métodos de ajuste algebraico a menudo producen una solución no iterativa única, mientras que los geométricos hacen uso de algunas restricciones no lineales mediante algoritmos iterativos (Muñoz-Pérez et al., 2014). En 2004, Li and Griffiths (2004) explica que el ajuste elipsoide se puede lograr de muchas maneras mediante la aplicación de algunas técnicas conocidas de ajuste de superficie limitada. Aun así, estas técnicas implican un método de optimización no lineal, que no puede garantizar una solución óptima y se detiene en un mínimo local. En contraste, Yu et al. (2009) menciona que los elipsoides generales no tienen una definición geométrica natural similar a las elipses o esferoides.

La mejora de la calidad y la resolución de la imagen es un objetivo con-

stante en las imágenes médicas, debido a la importancia crítica de estas imágenes para encontrar el diagnóstico y el tratamiento correctos para los pacientes. Esto no solo se refleja en la optimización de las técnicas de adquisición, por ejemplo, en el caso de la resonancia magnética, sino también en la etapa de posprocesamiento, con un interés cada vez mayor en algoritmos nuevos y mejorados. Las técnicas basadas en interpolación y spline son la estrategia más clásica y conocida para muestrear imágenes de RM (Pan et al., 2012; Thevenaz et al., 2000). Los métodos de interpolación lineal y bicúbica se han utilizado ampliamente.

Las técnicas de super-resolución (SR) ofrecen un mejor enfoque para mejorar la resolución (Wang et al., 2018; Yue et al., 2016). Los métodos de SR de imagen simple aplican SR en una sola imagen de baja resolución, que generalmente es un problema inverso mal condicionado. Los métodos basados en ejemplos se han vuelto populares como técnicas de super-resolución (Balure and Kini, 2015; Prince et al., 2019). Entre ellos, algunos explotan las similitudes internas de la imagen (Zheng et al., 2017) y otros aprenden patrones de mapeo entre imágenes de baja y alta resolución de conjuntos de datos externos (Trinh et al., 2014; Zhang et al., 2015). El enfoque más frecuente de una sola imagen ha sido el uso de una base de datos de imágenes de ejemplo, a partir de la cual los algoritmos aprenden a estimar la información de alta resolución perdida en la imagen de baja resolución.

Recientemente, una red neuronal convolucional de super-resolución (SRCNN) (Dong et al., 2016), que puede considerarse como un método basado en ejemplos, y su versión 3D (SRCNN3D) (Pham et al., 2017), han recibido gran atención debido a su capacidad para aprender un mapeo de extremo a extremo entre imágenes de baja y alta resolución, evitando así aprender de diccionarios o múltiples para modelar el espacio de alta resolución. Recientemente se presentó (Chen et al., 2018c) una red de super-resolución 3D densamente conectada para la mejora de RM del cerebro. Su particularidad es el uso de capas densas en lugar de capas convolucionales en su arquitectura. La alta complejidad esperada del método se mitiga porque es un modelo liviano con peso compartido y un número muy reducido de parámetros. Otro trabajo reciente presenta una combinación de una red densamente conectada con una red generativa antagónica (GAN), que parece lograr resultados prometedores (Chen et al., 2018b).

B.2.2 Fundamentos de la imagen

Dos formas principales de representación se distinguen por la forma en que se construye la imagen digitalizada, la representación de mapa de bits y la representación vectorial. La representación de mapa de bits de una imagen es una cuadrícula bidimensional de valores ubicados en celdas, mejor conocidos como píxeles, que se generan a intervalos regulares. A partir de aquí, el conjunto de píxeles representa los puntos infinitos de una imagen ideal.

Luego, los puntos de color se reducen a números para almacenamiento y tratamiento en la cuadrícula. Las imágenes se pueden representar en varias escalas de color. Los más comunes son la escala de grises y Red-Green-Blue (RGB). Por lo tanto, una imagen se puede representar mediante una matriz $N \times M$ y, según la cantidad de información que queramos almacenar, se emplean más dimensiones.

El interés de la imagen por RM en la medicina ha estado creciendo desde su inicio, por la capacidad de distinguir órganos, tejidos, huesos y otras estructuras dentro del cuerpo, y para proporcionar información tanto anatómica como funcional que ayude a controlar los procesos metabólicos en el mismo, además de no ser invasivo porque no implica irradiación del paciente. Los formatos de imagen mencionados anteriormente son la representación clásica de una imagen bidimensional. Sin embargo, en tareas médicas, es interesante analizar la distribución tridimensional de las diferentes partes del cuerpo. El voxel es una unidad cúbica que constituye la unidad mínima accionable de una matriz 3D y es, por lo tanto, el equivalente de los píxeles en una imagen 2D. En las imágenes de RM, cada vóxel contendrá un valor único, que estará representado por una unidad de medida establecida durante la creación de las imágenes. Por lo tanto, estamos tratando con un tipo de imagen 3D en escala de grises, pero con más precisión de lo habitual, ya que contiene detalles finos obtenidos durante el procedimiento de adquisición. Existen diferentes formatos para almacenar una imagen de RM. Como la mayoría de los conjuntos de datos se crearon utilizando el estándar NIfTI (Todd-Pokropek et al., 1992), que es el formato en el que nos centraremos.

La RM tuvo sus comienzos en 1973 cuando salieron las primeras publicaciones de Lauterbur y Mansfield (Lauterbur, 1973). La idea principal es que el paciente se coloca en un campo magnético uniforme, que es generado por un imán superconductor situado alrededor del paciente. Además, se sabe que el momento angular intrínseco ('espín') de un núcleo de hidrógeno en un campo magnético precesa sobre ese campo en la frecuencia de Larmor, que depende linealmente de la magnitud del campo mismo. Si se envía un pulso de radiofrecuencia a la frecuencia de Larmor, los espines del protón entrarán en un estado excitado por el efecto de la RM. Cuando el pulso se desvanece, volverán al estado no excitado, a su vez emitiendo un pulso de radiofrecuencia detectable desde el exterior. La imagen de RM se forma a partir de esta señal.

El objetivo de un estudio bien realizado de RM es obtener suficiente señal de vóxel en relación con el ruido, también llamado SNR, y medido como la relación entre la señal de vóxel y la desviación estándar del ruido, para observar tejidos de interés. En general, el ruido deriva de fluctuaciones aleatorias en la electrónica de la bobina receptora y la muestra. Existen otras fuentes de ruido, como el ruido de digitalización y el efecto fantasma pseudoaleatorio

debido a los giros en movimiento que pueden afectar la imagen reconstruida. Se ha demostrado que el ruido de RM se rige por una distribución específica, la distribución de Rice (Macovski, 1996). Este tipo de ruido es difícil de modelar y eliminar de las imágenes, por lo que se han publicado trabajos específicos y complejos para esta tarea. Otro aspecto crucial de la RM es su resolución. La dependencia de la señal con el volumen de la muestra es crucial para obtener imágenes de muy alta resolución, ya que la señal en cada vóxel está limitada por su volumen y la magnetización disponible para el campo dado. Además, el tiempo de adquisición también limita la señal que se puede recopilar, lo que afecta el ruido y la resolución. La posición (y otras características) de las bobinas también representan un papel.

B.2.3 Técnicas de aprendizaje

La presente investigación comprende muchas metodologías para mejorar algunos de los métodos existentes para la segmentación, el ajuste de puntos y la mejora de la imagen. Todos ellos tienen una cosa en común, la optimización de una función f para minimizar el error definido por esta función. La función f se llama formalmente *función objetivo*. En el aprendizaje automático, recibe el nombre de *función de coste* ya que siempre es necesario evaluar continuamente la calidad de un modelo de datos donde un mínimo implica un conjunto de parámetros posiblemente óptimos con un error óptimo. El objetivo es por tanto el hallazgo de un valor $x^\star$, que se llama la solución mínima global del problema de optimización.

Para resolver problemas, se usan comúnmente procedimientos iterativos para converger a una solución en un número finito de pasos. El método del *Descenso del Gradiente* (GD) es una técnica de optimización computacional frecuente para encontrar un mínimo de una función. El enfoque del descenso del gradiente comienza en un punto arbitrario, se mueve a lo largo del gradiente en ese punto hacia el siguiente punto, y este procedimiento se repite hasta converger en un punto estacionario. Una de las ventajas de GD es su convergencia. Se ha demostrado que en ciertas condiciones (funciones fuertemente convexas), el método de descenso de gradiente siempre converge a la solución óptima. Sin embargo, este método puede tardar en ejecutarse en grandes conjuntos de datos. Esto se debe a que una iteración del algoritmo de descenso de gradiente requiere una predicción para cada instancia en el conjunto de datos de entrenamiento. En aquellas situaciones en las que es necesario procesar grandes cantidades de datos, como en el aprendizaje profundo, una variación de descenso de gradiente llamada *Descenso del Gradiente Estocástico* es la opción adecuada que usar. En esta variación, la actualización se lleva a cabo para cada instancia de entrenamiento en lugar de hacerlo al final del lote de instancias.

Las redes neuronales artificiales eran originalmente un modelo computacional de sistemas nerviosos biológicos, compuesto por un conjunto de unidades

llamadas neuronas artificiales o nodos conectados entre sí. Hay dos fases en el modelado de problemas de optimización con redes neuronales: entrenamiento y pruebas. Tres factores muy importantes en la fase de entranamiento son *tasa de aprendizaje*, *retropropagación* y *función de coste*. Particularmente, la función de coste se refiere a una función ad hoc cuyo valor sólo puede ser aproximado. Una elección adecuada de la función de coste mejora el comportamiento de la red neuronal, y este es uno de los objetivos de esta investigación.

Una arquitectura de *aprendizaje profundo* es una red neuronal de varias capas con numerosas capas ocultas. Muchas de estas capas calculan funciones no lineales que, en el caso del análisis de imágenes, hacen que el sistema sea sensible a pequeños detalles e insensible a grandes variaciones como la iluminación, el fondo o el entorno de los objetos. La arquitectura más popular son las *Redes Neuronales Convolucionales* o CNNs, que utiliza varias capas convolucionales combinadas con otros tipos de capas. La idea general es que las capas convolucionales son capaces de extraer características sofisticadas de la imagen al tiempo que reducen la dimensionalidad de los datos procesados. La CNN ha demostrado un excelente rendimiento en el procesamiento de imágenes y videos. Centrándose en la super-resolución, una de las primeras redes profundas creadas para esta tarea fue SRCNN (Dong et al., 2016). La parte central de nuestra investigación trata de explotar los beneficios de la versión 3D de esta red profunda, y los nuevos modelos se han construido en torno a ella.

B.2.4 Espacios normados y medidas

La definición de las funciones de coste es esencial para determinar el objetivo del procedimiento de minimización. La norma euclídea es el método más utilizado para medir errores, aunque tiene sus ventajas y desventajas. La robustez se alcanza comúnmente con la norma absoluta, y se debe explorar el uso de otras normas ℓ_p.

El *espacio* ℓ_2 o *espacio euclídeo* en $\mathbb{R}^n$ es un espacio vectorial definido con las operaciones algebraicas habituales (suma y producto por un escalar) y con una norma definida por la suma de los cuadrados de los componentes del vector, que comúnmente se denomina norma ℓ_2 o norma euclídea. La norma ℓ_2 es la medida más utilizada porque es fácil de calcular y se entiende fácilmente. Además, se presta a operaciones algebraicas. Como aplicación al problema general de optimización de funciones, descubrimos que la media minimiza la suma de las distancias al cuadrado (norma ℓ_2) de cada muestra (Farebrother, 2013). La media aritmética tiene sus ventajas y desventajas. Las principales ventajas son que es la medida de tendencia central más utilizada y que usa toda la información disponible, devolviendo un valor único; representa el centro de gravedad en la distribución, siendo sensible a cualquier cambio en los datos. Por otro lado, los principales problemas están

asociados con la media son que es muy sensible a valores extremos, por lo que no se recomienda usarlo en distribuciones muy asimétricas, además de que no se puede calcular para datos cualitativos.

Análogamente, en lugar de usar los cuadrados de los componentes del vector, podríamos considerar el uso de sus valores absolutos. Esta formulación no mejora el peso de cada componente, y su definición es bastante simple. Comúnmente, a esta medida se le denomina norma ℓ_1. La principal ventaja de la norma ℓ_1 en comparación con la norma ℓ_2 es que es más robusta cuando los valores de datos están dispersos. Sin embargo, tiene la desventaja de que no es derivable. Además, la norma ℓ_2 es más utilizada porque tiene la valiosa propiedad de que es invariante bajo rotaciones, lo cual es útil para muchas aplicaciones (como el Análisis de Componentes Principales). Una medida relacionada estrechamente con la norma ℓ_1 es la mediana, que minimiza la suma de las desviaciones absolutas. A diferencia de la media, las principales ventajas de la mediana son que es una medida de posición robusta, los valores extremos no influyen ya que solo importan los valores centrales, y si hay datos censurados en la muestra, no es posible calcular la media; sin embargo, la mediana puede eventualmente calcularse. En el lado opuesto, hay algunas desventajas, como que no siempre es un valor único, no utiliza toda la información disponible en su cálculo, o que es difícil calcular si el número de observaciones es enorme.

Finalmente, y de manera más general, los espacios anteriores se pueden extender para cualquier número real fijo $p \geq 1$. El *espacio* ℓ_p en $\mathbb{R}^n$ es un espacio vectorial definido con las operaciones algebraicas habituales y con la norma ℓ_p definida por

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

Como p tiende hacia $+\infty$, la norma ℓ_p tiende hacia la función máximo, lo que significa que esencialmente estamos penalizando el argumento más grande de la norma, es decir, la mayor diferencia entre componentes si medimos distancias. Otro comportamiento importante es que para valores de p más grandes, las componentes vectoriales indeseables pueden esconderse debajo de la más grande, permitiendo la generación de modelos menos dispersos ya que las distancias calculadas no se ven tan afectadas por aquellos valores indeseables como con la norma ℓ_1, que da una penalización igual a todas las componentes.

B.3 Segmentación

La construcción de un modelo del fondo de una escena sigue siendo una tarea difícil en los sistemas de videovigilancia, en particular para cámaras en movimiento. El primer trabajo, en lo que a segmentación se refiere, presenta

un enfoque novedoso para construir un modelo de fondo panorámico basado en redes neuronales de aprendizaje competitivo y una interpolación lineal posterior por triangulación de Delaunay.

El enfoque puede manejar direcciones de cámara arbitrarias y zoom para un sistema de vigilancia basado en cámaras Pan-Tilt-Zoom (PTZ). Se utiliza una red neuronal competitiva para construir un modelo de fondo panorámico para la detección de objetos. Debido a la gran cantidad de información de entrada, se ha utilizado una gran cantidad de neuronas y los prototipos de neuronas se han organizado en un árbol cuádruple para evaluarlos rápidamente. Siguiendo la estrategia de (Luque-Baena et al., 2015), los vectores de entrada se dividen en dos secciones. La primera sección contiene la información posicional en la escena de video, mientras que la segunda sección contiene las características de color. De esta forma, solo se emplea la información posicional para determinar qué neurona es la ganadora. Por lo tanto, cada neurona representa el color promedio en un campo receptivo que es una región de una teselización de Voronoi del panorama completo.

Se han probado cuatro escenas para verificar la viabilidad del sistema, obteniendo resultados adecuados y exitosos. Además, se ha demostrado que nuestro enfoque supera a varios métodos competitivos. Es notable que el modelo propuesto utilice muy poco tiempo CPU para procesar cada fotograma de vídeo de entrada, lo que permite una integración fácil en un sistema de videovigilancia con cámara PTZ en tiempo real.

Como extensión de este trabajo, se presenta un enfoque parecido para construir un modelo de fondo panorámico basado en la red de gas neuronal. Además, también se propone un modelo de agregación de redes de gas neuronal. El gas neuronal se erige como una popular red neuronal de aprendizaje no supervisada, que se emplea comúnmente para fines de cuantificación de vectores (Dimitriadis et al., 2016; Antonakakis et al., 2016). Hay dos fases en el proceso de aprendizaje: primero, la fase de ordenación donde la tasa de aprendizaje experimenta una disminución lineal; y luego la fase de convergencia donde la tasa de aprendizaje permanece constante en un valor pequeño. Esto se debe a que la fase de ordenación es necesaria para la organización global de las neuronas de la red, y después de eso, el sistema se ejecuta durante un tiempo indefinidamente largo.

La cantidad de neuronas N requeridas para esta aplicación es de millones. Para reducir el tiempo requerido para actualizar la red, se utiliza un conjunto de redes de gas neuronal Q, cada una con neuronas N. Los miembros del conjunto operan en paralelo, de modo que cada red extrae aleatoriamente un vector de entrada del conjunto de datos de entrada independientemente de las otras redes. De esta forma, se entrena un total de neuronas QN, mientras que el tiempo de entrenamiento general puede ser cercano al de una sola red con N neuronas, siempre que estén disponibles al menos Q procesadores. Una vez que finaliza el entrenamiento, se construye una red unificada compuesta

de QN neuronas uniendo las neuronas que provienen de los Q miembros del conjunto. De esta manera, se obtiene una representación más precisa de la distribución de entrada con requisitos de tiempo de entrenamiento reducidos.

Después de probar el enfoque propuesto en varias secuencias interiores, los resultados demuestran que los métodos propuestos son efectivos y adecuados para su uso en aplicaciones de videovigilancia en tiempo real. Cabe señalar que la reducción en el número de neuronas (del 50% al 30% del total de píxeles) implica una ligera disminución de las métricas de rendimiento, aunque visualmente no es demasiado evidente. Además, los modelos de agregación ayudan a mejorar el modelado de fondo, aumentando las tasas de rendimiento.

Para terminar con esta parte, la segmentación de imágenes es un objetivo permanente en muchas aplicaciones médicas, con el fin de seguir mejorando la capacidad de diagnóstico y la predicción de curación. Para evaluar la tasa de curación de heridas en las úlceras del pie diabético, se miden algunos parámetros del área de la herida. Sin embargo, la heterogeneidad de las lesiones cutáneas diabéticas y el ruido presente en las imágenes capturadas por cámaras digitales hacen que la detección de heridas sea una tarea difícil.

En este último trabajo se propone un método basado en el aprendizaje profundo para la segmentación precisa de las regiones de la herida. En el método propuesto, las imágenes de entrada se procesan primero para eliminar artefactos y luego se alimentan a una red neuronal convolucional que produce un mapa de probabilidad. Finalmente, los mapas de probabilidad se procesan para extraer la región de la herida. Además, abordamos el problema de eliminar algunos falsos positivos.

La novedad es que diseñamos un modelo simplificado para procesar una amplia gama de heridas cutáneas diabéticas, incorporando un método de umbral adaptativo para eliminar artefactos. También realizamos un preprocesamiento para homogeneizar la entrada, y se incluyen algunas operaciones de postprocesamiento para eliminar artefactos ruidosos y seleccionar las regiones correctas de la herida en función de las probabilidades medias de las componentes conectadas. Para tener una mejor evaluación del rendimiento de nuestro modelo, también diseñamos un modelo de clasificación binaria SVM para la segmentación de imágenes, que se ha utilizado con éxito en problemas de clasificación y regresión, y comparamos los resultados con la CNN. Los experimentos muestran que el método propuesto puede lograr un alto rendimiento en términos de precisión de segmentación e índice de Dice.

B.4 Ajuste de curvas y superficies

Ajustar datos discretos obtenidos por dispositivos de adquisición de imágenes a una curva es una tarea común en muchos campos de la ciencia y la ingeniería. En particular, la parábola es una de las características de forma

más empleadas en aplicaciones de ingeniería eléctrica y telecomunicaciones. Las técnicas estándar de ajuste de curvas para resolver este problema implican la minimización de los errores al cuadrado. Sin embargo, la mayoría de estos procedimientos son sensibles al ruido. Aquí, proponemos un algoritmo basado en la minimización de errores absolutos acompañado de una normalización del vector directriz que conduce a una estabilidad mejorada del método. De esta manera, nuestra propuesta es sustancialmente resistente a las muestras ruidosas en el conjunto de datos de entrada.

La sensibilidad a las observaciones ruidosas se exacerba aún más por el hecho de que para muchos conjuntos de datos de entrada existen soluciones espurias en las que los algoritmos de ajuste pueden caer fácilmente. El objetivo es desarrollar un método para ajustar las parábolas a partir de un conjunto de puntos de muestra que sea más robusto, basado en la minimización de errores absolutos. La principal contribución de este trabajo es la propuesta de un algoritmo de adaptación que sea resistente tanto a las muestras anómalas, mediante la minimización de los errores absolutos, a la presencia de soluciones espúreas, como a la introducción de reinicios aleatorios en la búsqueda. Además, nuestra propuesta logra un buen equilibrio entre la calidad de la solución y la velocidad de ejecución. Los resultados experimentales demuestran el buen rendimiento del algoritmo en términos de velocidad y precisión en comparación con los enfoques anteriores, tanto para datos sintéticos como reales.

Los métodos de ajuste de elipses a menudo consideran las formas algebraicas y geométricas de la elipse. El trabajo presentado es esta tesis hace uso de un conjunto de ellos para proporcionar mejores resultados. El método propone una nueva parametrización de elipse basada en las coordenadas de ambos focos y la distancia entre ellos y cada punto de la elipse donde se aplica la norma euclídea. Además, un cierto número de subconjuntos se generan de manera uniforme sin reemplazo del conjunto de entrenamiento general, lo que permite estimar el centro de la distribución de manera sólida al emplear la mediana de ℓ_1 de cada foco estimado. También se propone una etapa adicional de postprocesamiento para filtrar el efecto de los malos ajustes.

Nuestra propuesta está motivada por la observación de que se pueden obtener varias soluciones alternativas del mismo algoritmo base cambiando ligeramente el conjunto de puntos de entrada. Por lo tanto, se puede calcular un ajuste mejorado combinando de manera robusta esas diferentes soluciones, de modo que los ajustes groseramente erróneos tengan un pequeño efecto en la solución final. Con el fin de lograr dicho objetivo, se propone una parametrización de la elipse que sea adecuada para el promedio robusto de las soluciones combinadas. La mediana se emplea para llevar a cabo dicho promedio. La razón detrás de esto es que la mediana es resistente a los valores atípicos, por lo que algunas soluciones completamente incorrectas no

arruinan la estimación general. El espacio de investigación que se aborda aquí es cuán preciso puede ser el promedio de conjuntos de ajustes de elipse, en comparación con los ajustes individuales de algoritmos clásicos que no consideran agregaciones. Dados los éxitos anteriores de los métodos de agregación cuando se aplican a otros campos de investigación, y la falta de este tipo de propuestas para el ajuste de elipse, se justifica la importancia de nuestra investigación.

La validez de nuestra propuesta se evalúa mediante estudios cuantitativos y cualitativos, en comparación con los algoritmos de ajuste de elipse más recientes. Para evaluar el rendimiento de este método, se consideraron cuatro medidas de error diferentes. Los resultados muestran que nuestra propuesta supera a todos sus competidores, especialmente cuando se presentan niveles más altos de valores atípicos. Se desarrollaron varias pruebas de datos sintéticos y reales y confirmaron dicho hallazgo.

Por último, la adaptación de superficies geométricas o algebraicas a datos 3D es un problema generalizado en muchos campos de la ciencia y la ingeniería. En particular, los elipsoides son algunas de las características más empleadas en gráficos de computador y calibraciones de sensores. También son útiles en el reconocimiento de patrones, la visión por computadora, la detección del cuerpo y el diseño de dispositivos electrónicos. Las técnicas de ajuste elipsoide estándar para resolver este problema implican la minimización de los errores cuadráticos. Sin embargo, la mayoría de estos procedimientos son sensibles al ruido. Aquí, proponemos un método basado en la minimización de errores absolutos.

La minimización de la suma de errores absolutos (también llamada minimización de la norma ℓ_1) se ha utilizado previamente para ajustar superficies en 3D (Flöry and Hofer, 2010). Los puntos base en la superficie no son necesarios en nuestro enfoque porque la forma analítica del elipsoide permite calcular la desviación de los puntos de muestra de la superficie sin puntos base. La función objetivo no es lineal, por lo que se utiliza una versión específicamente ajustada del descenso de gradiente en lugar de la programación lineal. El papel de la reformulación del valor absoluto por variables auxiliares en el método de Flöry y Hofer se cumple en nuestro método mediante la división de la nube de puntos en dos conjuntos: los puntos interiores y los puntos exteriores.

De lo anterior se puede decir que, aunque el tipo de superficies a ajustar y la metodología de nuestra propuesta difieren significativamente de Flöry and Hofer (2010), ambas aprovechan la robustez frente a los valores atípicos proporcionados por la optimización ℓ_1. Aunque nuestro algoritmo es iterativo, se utiliza un tamaño de paso adaptativo para lograr una convergencia más rápida. Esto conduce a una mejora sustancial en la solidez frente a datos atípicos. La propuesta se demuestra con varios ejemplos computacionales que comprenden datos sintéticos y datos reales de un escáner 3D y

una cámara estereoscópica.

B.5 Super-Resolución de RM

El proceso de adquisición de imágenes en el campo de la RM no siempre proporciona resultados de alta resolución que pueden ser útiles para un análisis clínico. Las técnicas de SR logran aumentar la resolución de la imagen, siendo especialmente efectivas aquellas basadas en ejemplos que determinan una correspondencia entre patrones de baja resolución y alta resolución. Las redes neuronales de aprendizaje profundo se han aplicado en los últimos años para estimar esta asociación con resultados muy competitivos. En este primer trabajo, una red neuronal convolucional 3D previamente definida (Pham et al., 2017) (SRCNN3D), se combina con un mecanismo de desplazamiento espacial regular con el objetivo de mejorar sustancialmente la calidad de la imagen resultante. Por lo tanto, la combinación de la salida de red después de probar varias imágenes de entrada desplazadas de forma regular generar una imagen de alta resolución mejorada.

Esta propuesta híbrida se ha aplicado satisfactoriamente a diversos conjuntos de datos de imágenes de RM cerebrales y se ha comparado con varios algoritmos de vanguardia de SR. Además, se ha incluido en la comparación otra propuesta publicada previamente de nuestro grupo de investigación (Thurnhofer-Hemsi et al., 2018b). Utilizando las métricas PSNR, SSIM y BC, los resultados obtenidos en diferentes imágenes de RM muestran una mejora considerable tanto en la imagen restaurada como en la imagen residual sin un aumento excesivo en el tiempo de cálculo.

Por otro lado, las técnicas de mejora de imagen son frecuentes, pero hay pocas propuestas que aborden el aumento de la resolución espacial y la eliminación de ruido al mismo tiempo. En el siguiente trabajo se propone dicho tipo de algoritmo, siendo una versión modificada de la Median Filter Transform (López-Rubio, 2016). Se tesela el espacio de la imagen 3D en paralelepípedos, de modo que se aplica un filtro de la mediana en cada paralelepípedo. Los resultados obtenidos de varias de estas teselaciones se combinan mediante el cálculo de la mediana de dichos resultados. Las propiedades de convergencia de la propuesta se prueban formalmente. Se reportan resultados experimentales con imágenes sintéticas y reales, donde nuestro enfoque supera a sus competidores para niveles de ruido altos. Además, se demuestra que nuestro algoritmo no genera ninguna alucinación.

El algoritmo proporciona una implementación simple, y no hace suposiciones sobre el tipo de ruido o el nivel de ruido, lo que facilita su uso para diferentes técnicas de imagen. No se hacen suposiciones sobre los parámetros del sistema de imagen, como la función de dispersión de puntos, y, por lo tanto, el procedimiento no se ve afectado por errores en las estimaciones de esos parámetros. Además, se permiten escalas de zoom fraccionarias ar-

bitrarias sin la ayuda de un algoritmo de interpolación adicional. Como nuestra propuesta no necesita imágenes adicionales, la aplicabilidad del algoritmo está garantizada en cualquier contexto. Esto también elimina la necesidad de tiempos de adquisición más largos y las dificultades adicionales asociadas con los problemas de registro, la necesidad de tener un conjunto de datos externo, o la dependencia de sus características.

Para terminar, el estado del arte reúne una gran variedad de métodos para SR, entre los cuales el aprendizaje profundo se ha vuelto muy popular durante los últimos años. La mayoría de los métodos de aprendizaje profundo de SR se basan en la minimización de los residuos mediante el uso de capas de pérdida euclídeas. En los últimos tiempos, el uso de métodos de norma ℓ_p en algoritmos de optimización ha llamado la atención para varias aplicaciones. Se han utilizado para los clasificadores binarios de aprendizaje automático (Grove et al., 2001; Gentile, 2003), y en el marco del control óptimo (Blueschke and Savin, 2017). Una de las ventajas de usar una norma ℓ_p, con $1 < p < 2$, es que puede reducir el efecto de los valores atípicos en un problema de minimización. El ruido y los artefactos en las imágenes del conjunto de entrenamiento para la super-resolución de RM son valores atípicos que deben evitarse para que un algoritmo de SR proporcione soluciones realistas de alta calidad.

En esta tesis, se propone la norma ℓ_p con $1 < p < 2$ como la función de pérdida para las capas neurales de las redes neuronales convolucionales de super-resolución, y se estudian los valores más adecuados de p. Este primer método se implementó usando una CNN tridimensional, y se estudiaron los valores más adecuados de p para determinar el ajuste más robusto. La metodología propuesta fue entrenada y probada con conjuntos de imágenes de RM estructurales, mostrando mejores resultados cuantitativamente, en términos de PSNR y SSIM, y las imágenes residuales restauradas y calculadas mostraron mejores resultados.

Por último, se han desarrollado métodos de optimización multiobjetivo para el caso en el que se persigue más de un objetivo en el proceso de optimización. Por ejemplo, la mejora de la imagen tiene como objetivo reducir el ruido mientras se conservan los pequeños detalles. Por lo general, estos dos objetivos chocan, ya que la reducción de ruido a menudo se logra suavizando la imagen y eliminando así los detalles. La optimización multiobjetivo permite combinar dos o más funciones de pérdida y, según los métodos y parámetros utilizados, dar más relevancia al objetivo más relevante en cada caso. La optimización multiobjetivo se basa en la premisa de que la mejora de uno de los objetivos puede conducir al deterioro de otro objetivo. Por lo tanto, una solución globalmente óptima no es posible, por lo que se requiere una búsqueda en el frente de Pareto. La escalarización es otro enfoque para la optimización multiobjetivo, que se basa en la optimización de una función escalar que combina las múltiples funciones objetivo del prob-

lema original. Este segundo método se centra en una particularización de la escalarización Pascoletti-Serafini (Eichfelder, 2009), que garantiza generar soluciones adecuadas simplemente seleccionando los pesos adecuados.

A la luz de lo anterior, el objetivo de este trabajo es obtener una red de aprendizaje profundo mejorada mediante el uso de un enfoque de optimización multiobjetivo con dos funciones de pérdida basadas en la norma ℓ_p y encontrar los parámetros de la nueva función de coste que brinden los mejores resultados para sus aplicación a imágenes 3D de RM. La combinación de dos funciones de coste alternativas es un enfoque novedoso para el entrenamiento de redes neuronales convolucionales profundas ya que el enfoque estándar para el aprendizaje profundo se basa en la minimización por el descenso de gradiente estocástico de una sola función de pérdida. Las capas de pérdida propuestas se entrenaron con las redes SRCNN3D y DCSRN y se probaron con dos conjuntos de datos de RM, y luego se compararon con la función de pérdida euclídea tradicional. Los resultados experimentales muestran diferencias significativas, mientras que las imágenes residuales muestran detalles más refinados.

B.6 Conclusiones

Esta tesis cubre varios temas del procesamiento de imágenes mediante el uso de varias técnicas de inteligencia computacional. Se han desarrollado metamodelos generales, así como aplicaciones específicas, para su aplicación en tres áreas diferentes: videovigilancia, ajuste geométrico e imagen médica. Para ese fin, se han utilizado técnicas de optimización y redes neuronales artificiales basadas en las normas ell_1, ℓ_2 y, más en general, ℓ_p para mejorar los métodos de vanguardia. Hemos seguido un flujo de trabajo típico en el procesamiento de imágenes, que comprende la segmentación de la imagen, el ajuste posterior de alguna forma geométrica y la mejora de las propiedades cualitativas de la imagen. Dicho esto, esta tesis se ha dividido en tres partes principales, que también coinciden en mayor medida con los diferentes campos de aplicación de las propuestas.

La primera parte se centra en la segmentación del fondo y el primer plano de las imágenes. Primero, se presentaron dos trabajos que modelan el fondo panorámico grabado por una cámara PTZ. Por otro lado, se describió un sistema de segmentación para heridas cutáneas diabéticas. La particularidad de los trabajos presentados en esta parte es que la norma ℓ_2 y la media se han utilizado como las técnicas principales para implementar los métodos. La segunda parte recopila tres métodos de ajuste de puntos para formas geométricas 2D y 3D. En estos casos, se necesitaba más precisión, por lo que la norma ℓ_1 y la mediana han sido las métricas empleadas en su mayoría. La tercera y última parte se centra en el problema de la super-resolución, y más específicamente, en su aplicación en las imágenes de RM del cerebro. El

requisito de altos niveles de precisión en el proceso de mejora de imágenes de imágenes médicas es una máxima para evitar un mal diagnóstico. Los Capítulos 12-15 propusieron cuatro métodos SR aplicados a la mejora de las imágenes cerebrales, en donde se ha ido variando el uso de la norma ℓ_p, con $p \in [1, 2]$, para ir generando métodos cada vez más precisos.

Resumiendo toda la investigación realizada en esta tesis, podemos concluir que el proceso de optimización es la clave para obtener un rendimiento u otro, dependiendo de los objetivos del trabajo. Comenzando por el uso de técnicas clásicas de inteligencia computacional como el descenso de gradiente o redes neuronales simples (Capítulos 6, 7, 9-11, y 13), la aparición del aprendizaje profundo en el medio de esta investigación ha introducido un nuevo desafío para ser abordado (Capítulos 8, 12, 14, and 15). Los estudios realizados fueron variados, pero con la misma pregunta en mente, cómo se puede mejorar la optimización. Por último, mostramos que las técnicas propuestas podrían aplicarse a cualquier campo de procesamiento de imágenes, lo que indica la amplitud y aplicabilidad a esta área de la Ciencia.

Algunas de las futuras líneas de investigación son las siguientes:

- Diseñar nuevos algoritmos para la segmentación en escenarios difíciles específicos; desarrollar métodos de ajuste para otras formas geométricas de interés, como paraboloides o cualquier otro; adaptar y aplicar los metamodelos SR propuestos para cualquier tipo de imagen, tanto RGB como médica. La norma ℓ_2 y la media parecen ser insuficientes para los problemas de optimización más difíciles.
- Mejorar las redes profundas de última generación mediante el uso de funciones de coste adaptadas, ya sea con normas ℓ_p o cualquier otra medida de error como SSIM.
- Desarrollar métodos de optimización multi-objetivo basados en más de dos normas ℓ_p.
- Estudiar, desde un punto de vista teórico, la convergencia y la eficacia de los métodos de optimización utilizando la norma ℓ_p.

Bibliography

Abramovich, F., Benjamini, Y., Donoho, D. L., and Johnstone, I. M. (2006). Adapting to unknown sparsity by controlling the false discovery rate. *Annals of Statistics*, 34(2):584–653.

Ahn, S. J., Rauh, W., and Warnecke, H. J. (2001). Least-squares orthogonal distances fitting of circle, sphere, elipse, hyperbola, and parabola. *Pattern Recognition*, 34(12):2283–2303.

Amasyali, M. F. (2019). Improved Space Forest: A Meta Ensemble Method. *IEEE Transactions on Cybernetics*, 49(3):816–826.

Anand, C. S. and Sahambi, J. S. (2010). Wavelet domain non-linear filtering for MRI denoising. *Magnetic Resonance Imaging*, 28(6):842–861.

Anand, S. and Cheresh, D. A. (2011). MicroRNA-mediated regulation of the angiogenic switch. *Current Opinion in Hematology*, 18(3):171–176.

Antonakakis, M., Dimitriadis, S. I., Papanicolaou, A. C., Zouridakis, G., and Zervakis, M. (2016). Improving the detection of mtbi via complexity analysis in resting - State magnetoencephalography. In *IST 2016 - 2016 IEEE International Conference on Imaging Systems and Techniques, Proceedings*, pages 156–160.

Arratia, R. and Gordon, L. (1989). Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology*, 51(1):125–131.

Asha Gnana Priya, H., Anitha, J., and Poonima Jacinth, J. (2018). Identification of Melanoma in Dermoscopy Images Using Image Processing Algorithms. In *2018 International Conference on Control, Power, Communication and Computing Technologies, ICCPCCT 2018*, pages 553–557.

Azzari, P. and Bevilacqua, A. (2006). Joint spatial and tonal mosaic alignment for motion detection with PTZ camera. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4142 LNCS:764–775.

Babu, K., Subudhi, A., and Sabut, S. (2018). Segmentation of Diabetic Wound by Multidimensional Clustering for Quantitative Assessment of Healing Process. *Current Medical Imaging Reviews*, 14(1):71–76.

Balure, C. S. and Kini, M. R. (2015). A Survey-Super Resolution Techniques for Multiple, Single, and Stereo Images. In *Proceedings - 2014 5th International Symposium on Electronic System Design, ISED 2014*, pages 215–216.

Bar-Yossef, Z., Jayram, T. S., Kumar, R., and Sivakumar, D. (2004). An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732.

Barata, C., Celebi, M. E., and Marques, J. S. (2015). Improving dermoscopy image classification using color constancy. *IEEE Journal of Biomedical and Health Informatics*, 19(3):1146–1152.

Bengtsson, H., Jacobson, A., and Riedy, J. (2015). R.matlab: Read and Write MAT Files and Call MATLAB from Within R.

Benko, P., Kós, G., Várady, T., Andor, L., and Martin, R. (2002). Constrained fitting in reverse engineering. *Computer Aided Geometric Design*, 19(3):173–205.

Bevilacqua, A. and Azzari, P. (2006). High-quality real time motion detection using PTZ cameras. *Proceedings - IEEE International Conference on Video and Signal Based Surveillance 2006, AVSS 2006*, pages 1–6.

Bhattacharyya, A. (1946). On a Measure of Divergence between Two Multinomial Populations. *Sankhyā: The Indian Journal of Statistics*, 7(4):401–406.

Bidgood, W. D., Horii, S. C., Prior, F. W., and Van Syckle, D. E. (1997). Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. *Journal of the American Medical Informatics Association*, 4(3):199–212.

Bioucas-Dias, J. M. and Valadão, G. (2007). Phase unwrapping via graph cuts. *IEEE Transactions on Image Processing*, 16(3):698–709.

Blane, M. M. (2000). The 3L algorithm for fitting implicit polynomial curves and surfaces to data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):298–313.

Blueschke, D. and Savin, I. (2017). No such thing as a perfect hammer: comparing different objective function specifications for optimal control. *Central European Journal of Operations Research*, 25(2):377–392.

Bookstein, F. L. (1979). Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9(1):56–71.

Borges, A. F. (2017). Analysis of wave velocity anisotropy of rocks using ellipse fitting. *International Journal of Rock Mechanics and Mining Sciences*, 96:23–33.

Boult, T. E., Gao, X., Micheals, R., and Eckmann, M. (2004). Omnidirectional visual surveillance. *Image and Vision Computing*, 22(7):515–534.

Bouwmans, T., Silva, C., Marghes, C., Zitouni, M. S., Bhaskar, H., and Frelicot, C. (2018). On the role and the importance of features for background modeling and foreground detection. *Computer Science Review*, 28:26–91.

Bowman, V. J. (1976). On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-Criteria Objectives. In Thiriez, H. and Zionts, S., editors, *Multiple Criteria Decision Making*, pages 76–86, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bramwell, M. C. and Kreyszig, E. (1979). Introductory Functional Analysis with Applications. *The Mathematical Gazette*, 63(424):137.

Branham, R. L., J. (1982). Alternatives to least squares. *The Astronomical Journal*, 87:928.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Brown, R. W., Cheng, Y. C. N., Haacke, E. M., Thompson, M. R., and Venkatesan, R. (2014). *Magnetic Resonance Imaging: Physical Principles and Sequence Design: Second Edition*, volume 9780471720850. John Wiley & Sons Ltd, Chichester, UK.

Bui, D. T., Tsangaratos, P., Ngo, P. T. T., Pham, T. D., and Pham, B. T. (2019). Flash flood susceptibility modeling using an optimized fuzzy rule based feature selection technique and tree based ensemble methods. *Science of the Total Environment*, 668:1038–1054.

Butler, B. P., Forbes, A. B., and Harris, P. M. (1994). Algorithms for geometric tolerance assessment. Technical report, National Physical Lab., Teddington (United Kingdom). Div. of Information Technology and Computing.

Calafiore, G. (2002). Approximation of n-dimensional data using spherical and ellipsoidal primitives. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 32(2):269–278.

Calafiore, G. C. (2000). Outliers robustness in multivariate orthogonal regression. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 30(6):674–679.

Cao, X. and Shrikhande, N. (2002). Quadric surface fitting for sparse range data. In *Systems, Man, and Cybernetics, 1991. 'Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on*, volume 1, pages 123–128.

Cermeño, E., Pérez, A., and Sigüenza, J. A. (2018). Intelligent video surveillance beyond robust background modeling. *Expert Systems with Applications*, 91:138–149.

Chang, Q., Hung, K. W., and Jiang, J. (2018). Deep learning based image Super-resolution for nonlinear lens distortions. *Neurocomputing*, 275:969–982.

Chen, C. C., Yao, Y., Drira, A., Koschan, A., and Abidi, M. (2009). Cooperative mapping of multiple PTZ cameras in automated surveillance systems. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, volume 2009 IEEE, pages 1078–1084. IEEE.

Chen, C. H., Yao, Y., Page, D., Abidi, B., Koschan, A., and Abidi, M. (2008). Heterogeneous fusion of omnidirectional and PTZ cameras for multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1052–1063.

Chen, G., St-Charles, P. L., Bouachir, W., Bilodeau, G. A., and Bergevin, R. (2015). Reproducible evaluation of Pan-Tilt-Zoom tracking. *Proceedings - International Conference on Image Processing, ICIP*, 2015-Decem:2055–2059.

Chen, L., Bentley, P., Mori, K., Misawa, K., Fujiwara, M., and Rueckert, D. (2018a). DRINet for Medical Image Segmentation. *IEEE Transactions on Medical Imaging*, 37(11):2453–2462.

Chen, S., Xia, R., Zhao, J., Chen, Y., and Hu, M. (2017a). A hybrid method for ellipse detection in industrial images. *Pattern Recognition*, 68:82–98.

Chen, X., Xu, F., and Ye, Y. (2010). Lower bound theory of nonzero entries in solutions of l2-l p minimization. *SIAM Journal on Scientific Computing*, 32(5):2832–2852.

Chen, Y., Shi, F., Christodoulou, A. G., Xie, Y., Zhou, Z., and Li, D. (2018b). Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network. In

Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 11070 LNCS, pages 91–99. Springer.

Chen, Y., So, H. C., and Kuruoglu, E. E. (2016). Variance analysis of unbiased least lp-norm estimator in non-Gaussian noise. *Signal Processing*, 122:190–203.

Chen, Y., So, H. C., Kuruoglu, E. E., and Yang, X. L. (2017b). Variance analysis of unbiased complex-valued lp-norm minimizer. *Signal Processing*, 135:17–25.

Chen, Y., Xie, Y., Zhou, Z., Shi, F., Christodoulou, A. G., and Li, D. (2018c). Brain MRI super resolution using 3D deep densely connected neural networks. In *Proceedings - International Symposium on Biomedical Imaging*, volume 2018-April, pages 739–742.

Chen, Y. H. and Liu, C. Y. (1999). Quadric surface extraction using genetic algorithms. *CAD Computer Aided Design*, 31(2):101–110.

Cheng, G., Zhou, P., and Han, J. (2016). Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415.

Cheng, J., Zhang, G., Caraffini, F., and Neri, F. (2015). Multicriteria adaptive differential evolution for global numerical optimization. *Integrated Computer-Aided Engineering*, 22(2):103–117.

Chenouri, S., Liang, J., and Small, C. G. (2015). Robust dimension reduction. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(1):63–69.

Chernov, N. and Lesort, C. (2004). Statistical efficiency of curve fitting algorithms. *Computational Statistics and Data Analysis*, 47(4):713–728.

Chernov, N. and Lesort, C. (2005). Least squares fitting of circles. *Journal of Mathematical Imaging and Vision*, 23(3):239–252.

Cheung, S.-c. S. and Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. In Panchanathan, S. and Vasudev, B., editors, *Visual Communications and Image Processing 2004*, volume 5308, page 881.

Chin, K. S., Trucco, E., Tan, L., and Wilson, P. J. (2013). Automatic fovea location in retinal images using anatomical priors and vessel density. *Pattern Recognition Letters*, 34(10):1152–1158.

Chojnacki, W., Brooks, M. J., and Vanel, A. (2000). On the fitting of surfaces to data with covariances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1294–1303.

Chong, J. J. (2020). Deep-Learning Super-Resolution MRI: Getting Something From Nothing. *Journal of Magnetic Resonance Imaging*, 51(4):1140–1141.

Chung, P. Y. and Ng, G. Y. (2012). Comparison between an accelerometer and a three-dimensional motion analysis system for the detection of movement. *Physiotherapy (United Kingdom)*, 98(3):256–259.

Collings, S., Kozera, R., and Noakes, L. (2009). Robust surface fitting from Two views using restricted correspondence. *Journal of Mathematical Imaging and Vision*, 34(2):200–221.

Comaniciu, D. and Meer, P. (1997). Robust analysis of feature spaces: Color image segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 750–755. IEEE Comput. Soc.

Comport, A. I., Marchand, E., Pressigout, M., and Chaumette, F. (2006). Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628.

Conti, C., Romani, L., and Schenone, D. (2018). Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform. *Pattern Recognition*, 74:64–76.

Coupé, P., Hellier, P., Prima, S., Kervrann, C., and Barillot, C. (2008). 3D wavelet subbands mixing for image denoising. *International Journal of Biomedical Imaging*, 2008(1):1:1—-1:11.

Coupé, P., Manjón, J. V., Fonov, V., Pruessner, J., Robles, M., and Collins, D. L. (2011). Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2):940–954.

Coupe, P., Yger, P., Prima, S., Hellier, P., Kervrann, C., and Barillot, C. (2008). An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. *IEEE Transactions on Medical Imaging*, 27(4):425–441.

Cross, G. and Zisserman, A. (1998). Quadric reconstruction from dual-space geometry. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 25–31.

Cui, C., Thurnhofer-Hemsi, K., Soroushmehr, R., Mishra, A., Gryak, J., Dominguez, E., Najarian, K., and Lopez-Rubio, E. (2019). Diabetic Wound Segmentation using Convolutional Neural Networks. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 1002–1005. IEEE.

Datar, M., Gionis, A., Indyk, P., and Motwani, R. (2002). Maintaining stream statistics over sliding windows (extended abstract). *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 06-08-Janu(6):635–644.

Daubechies, I., Devore, R., Fornasier, M., and Güntürk, C. S. (2010). Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38.

De La Fraga, L. G., Silva, I. V., and Cruz-Cortes, N. (2007). Euclidean distance fit of ellipses with a genetic algorithm. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4448 LNCS, pages 359–366.

de Vieilleville, F. and Lachaud, J. O. (2009). Comparison and improvement of tangent estimators on digital curves. *Pattern Recognition*, 42(8):1693–1707.

Derrac, J., García, S., Molina, D., and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18.

Devassy, B. M., Yildirim-Yayilgan, S., and Hardeberg, J. Y. (2019). The Impact of Replacing Complex Hand-Crafted Features with Standard Features for Melanoma Classification Using Both Hand-Crafted and Deep Features. *Advances in Intelligent Systems and Computing*, 868:150–159.

Dimitriadis, S., Sun, Y., Laskaris, N., Thakor, N., and Bezerianos, A. (2016). Revealing Cross-Frequency Causal Interactions During a Mental Arithmetic Task Through Symbolic Transfer Entropy: A Novel Vector-Quantization Approach. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(10):1017–1028.

Ding, C., Morye, A. A., Farrell, J. A., and Roy-Chowdhury, A. K. (2012). Opportunistic sensing in a distributed PTZ camera network. *2012 6th International Conference on Distributed Smart Cameras, ICDSC 2012*, 21(7):3282–3295.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming, Series B*, 91(2):201–213.

Dong, C., Loy, C. C., He, K., and Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307.

Dong, W., Zhang, L., Lukac, R., and Shi, G. (2013). Sparse representation based image interpolation with nonlocal autoregressive modeling. *IEEE Transactions on Image Processing*, 22(4):1382–1394.

Dong, W., Zhang, L., Shi, G., and Wu, X. (2011). Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857.

Dougalis, V. A. and Kampanis, N. A. (1996). Finite element methods for the parabolic equation with interfaces. *Journal of Computational Acoustics*, 4(1):55–88.

Dougalis, V. A., Kampanis, N. A., Sturm, F., and Zouraris, G. E. (2008). Numerical solution of the parabolic equation in range–dependent waveguides. In *Effective Computational Methods for Wave Propagation*, pages 175–207. CRC Press/Taylor & Francis.

Du, X. and He, Y. (2019). Gradient-Guided Convolutional Neural Network for MRI Image Super-Resolution. *Applied Sciences*, 9(22):4874.

Durrant, R. J. and Kabán, A. (2015). Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions. *Machine Learning*, 99(2):257–286.

Eberly, D. (2015). Geometric Tools, LLC.

Ehrgott, M. and Ryan, D. M. (2002). Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3):139–150.

Eichfelder, G. (2009). An Adaptive Scalarization Method in Multiobjective Optimization. *SIAM Journal on Optimization*, 19(4):1694–1718.

Eliazar, I. (2020). *Order Statistics*. Wiley, Hoboken, N.J., 3rd ed. edition.

Emmerich, M. T. and Deutz, A. H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*, 17(3):585–609.

Fang, J. and Liu, Z. (2015). A new inclination error calibration method of motion table based on accelerometers. *IEEE Transactions on Instrumentation and Measurement*, 64(2):487–493.

Farebrother, R. W. (2013). *L_1-Norm and L_∞-Norm Estimation. An Introduction to the Least Absolute Residuals, the Minimax Absolute Residual and Related Fitting Procedures.* SpringerBriefs in Statistics. Springer Berlin Heidelberg, Berlin, Heidelberg.

Farsiu, S., Robinson, D., Elad, M., and Milanfar, P. (2004a). Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2):47–57.

Farsiu, S., Robinson, M. D., Elad, M., and Milanfar, P. (2004b). Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344.

Feng, W., Liu, S., Liu, S., and Yang, S. (2013). A calibration method of three-axis magnetic sensor based on ellipsoid fitting. *Journal of Information and Computational Science*, 10(6):1551–1558.

Figueiredo, M. A., Bioucas-Dias, J. M., and Nowak, R. D. (2007). Majorization-minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 16(12):2980–2991.

Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., Van Der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., Montillo, A., Makris, N., Rosen, B., and Dale, A. M. (2002). Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–355.

Fitzgibbon, A. and Fisher, R. (2013). A Buyer's Guide to Conic Fitting. *British Machine Vision Conference*, pages 51.1–51.10.

Fitzgibbon, A., Pilu, M., and Fisher, R. B. (1999). Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480.

Fitzgibbon, A. W., Pilu, M., and Fisher, R. B. (1996). Direct least squares fitting of ellipses. *Proceedings - International Conference on Pattern Recognition*, 1(5):253–257.

Flöry, S. and Hofer, M. (2010). Surface fitting and registration of point clouds using approximations of the unsigned distance function. *Computer Aided Geometric Design*, 27(1):60–77.

Freeman, W. T., Jones, T. R., and Pasztor, E. C. (2002). Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65.

Freund, Y. and Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, volume 1, pages 148–156.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28(2):337–407.

Frisken, S. and Perry, R. (2005). Simple and Efficient Traversal Methods for Quadtrees and Octrees. *Graphics Tools—The jgt Editors' Choice*, 7:3–14.

G., W., Mason, J. C., and Cox, M. G. (1988). Algorithms for Approximation. *Mathematics of Computation*, 51(183):386.

Gander, W., Golub, G. H., and Strebel, R. (1994). Least-squares fitting of circles and ellipses. *Bit*, 34(4):558–578.

Gao, L. (2001). Numerical algorithms for nonlinear Lp-norm problem and its extreme case. *Journal of Computational and Applied Mathematics*, 129(1-2):139–150.

Gao, Z., Chung, J., Abdelrazek, M., Leung, S., Hau, W. K., Xian, Z., Zhang, H., and Li, S. (2020a). Privileged Modality Distillation for Vessel Border Detection in Intracoronary Imaging. *IEEE Transactions on Medical Imaging*, 39(5):1524–1534.

Gao, Z., Wang, X., Sun, S., Wu, D., Bai, J., Yin, Y., Liu, X., Zhang, H., and de Albuquerque, V. H. C. (2020b). Learning physical properties in complex visual scenes: An intelligent machine for perceiving blood flow dynamics from static CT angiography imaging. *Neural Networks*, 123:82–93.

Gao, Z., Wu, S., Liu, Z., Luo, J., Zhang, H., Gong, M., and Li, S. (2019). Learning the implicit strain reconstruction in ultrasound elastography using privileged information. *Medical Image Analysis*, 58:101534.

García, S., Fernández, A., Luengo, J., and Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064.

García-Escudero, L. A., Mayo-Iscar, A., and Sánchez-Gutiérrez, C. I. (2017). Fitting parabolas in noisy images. *Computational Statistics and Data Analysis*, 112:80–87.

Gass, S. and Saaty, T. (1955). The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1-2):39–45.

Ge, Z., Sahiner, B., Chan, H. P., Hadjiiski, L. M., Cascade, P. N., Bogot, N., Kazerooni, E. A., Wei, J., and Zhou, C. (2005). Computer-aided detection of lung nodules: False positive reduction using a 3D gradient field method and 3D ellipsoid fitting. *Medical Physics*, 32(8):2443–2454.

Gentile, C. (2003). The Robustness of the p-Norm Algorithms. *Machine Learning*, 53(3):265–299.

Gersho, A. (1992). Competitive Learning and Soft Competition for Vector Quantizer Design. *IEEE Transactions on Signal Processing*, 40(2):294–309.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Giagkiozis, I. and Fleming, P. J. (2015). Methods for multi-objective optimization: An analysis. *Information Sciences*, 293:338–350.

Gietzelt, M., Wolf, K. H., Marschollek, M., and Haux, R. (2013). Performance comparison of accelerometer calibration algorithms based on 3D-ellipsoid fitting methods. *Computer Methods and Programs in Biomedicine*, 111(1):62–71.

Gilbert, K. E. (1993). A fast Green's function method for one-way sound propagation in the atmosphere. *Journal of the Acoustical Society of America*, 94(4):2343–2352.

Gilbert, K. E. and White, M. J. (1989). Application Of The Parabolic Equation To Sound Propagation In A Refracting Atmosphere. *Journal of the Acoustical Society of America*, 85(2):630–637.

Golub, G. H., Hoffman, A., and Stewart, G. W. (1987). A generalization of the Eckart-Young-Mirsky matrix approximation theorem. *Linear Algebra and Its Applications*, 88-89(C):317–327.

Gómez-Silva, M. J., Izquierdo, E., De La Escalera, A., and Armingol, J. M. (2019). Transferring learning from multi-person tracking to person re-identification. *Integrated Computer-Aided Engineering*, 26(4):329–344.

Gong, X. H. (2011). Chebyshev scalarization of solutions to the vector equilibrium problems. *Journal of Global Optimization*, 49(4):607–622.

Gontar, A., Tronnolone, H., Binder, B. J., and Bottema, M. J. (2018). Characterising shape patterns using features derived from best-fitting ellipsoids. *Pattern Recognition*, 83:365–374.

González-Villanueva, L., Cagnoni, S., and Ascari, L. (2013). Design of a wearable sensing system for human motion monitoring in physical rehabilitation. *Sensors (Switzerland)*, 13(6):7735–7755.

Gotardo, P. F. U., Bellon, O. R. P., Boyer, K. L., and Silva, L. (2004). Range image segmentation into planar and quadric surfaces using an improved robust estimator and genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(6):2303–2316.

Graepel, T. (2016). AlphaGo - Mastering the game of go with deep neural networks and tree search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9852 LNAI:XXI.

Grammalidis, N. and Strintzis, M. G. (2000). Head detection and tracking by 2-D and 3-D ellipsoid fitting. In *Proceedings of Computer Graphics International Conference, CGI*, pages 221–226.

Greenspan, H. (2008). Super-Resolution in Medical Imaging. *The Computer Journal*, 52(1):43–63.

Griffin, G., Holub, a., and Perona, P. (2007). Caltech-256 object category dataset. *Caltech mimeo*, 11(1):20.

Grivon, D., Vezzetti, E., and Violante, M. G. (2013). Development of an innovative low-cost MARG sensors alignment and distortion compensation methodology for 3D scanning applications. *Robotics and Autonomous Systems*, 61(12):1710–1716.

Grove, A. J., Littlestone, N., and Schuurmans, D. (2001). General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210.

Halir, R. and Flusser, J. (1998). Numerically Stable Direct Least Squares Fitting Of Ellipses. In *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*, volume 98, pages 125–132, Czech Republic. Department of Software Engineering, Charles University.

Han, J., Kim, J. S., Chung, C. K., and Park, K. S. (2007). Evaluation of smoothing in an iterative lp-norm minimization algorithm for surface-based source localization of MEG. *Physics in Medicine and Biology*, 52(16):4791–4803.

Harker, M., O'Leary, P., and Zsombor-Murray, P. (2008). Direct type-specific conic fitting and eigenvalue bias correction. *Image and Vision Computing*, 26(3):372–381.

Harris, J. W. and Stocker, H. (1998). *Handbook of Mathematics and Computational Science*. Springer-Verlag.

Hathaway, R. J., Bezdek, J. C., and Hu, Y. (2000). Generalized fuzzy c-means clustering strategies using Lp norm distances. *IEEE Transactions on Fuzzy Systems*, 8(5):576–582.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2016-Decem, pages 770–778. IEEE.

He, R., Hu, B. G., Zheng, W. S., and Kong, X. W. (2011). Robust principal component analysis based on maximum correntropy criterion. *IEEE Transactions on Image Processing*, 20(6):1485–1494.

Helzer, A., Barzohar, M., and Malah, D. (2004). Stable fitting of 2D curves and 3D surfaces by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1283–1294.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Hofmann, T. and Buhmann, J. M. (1998). Competitive Learning algorithms for robust vector quantization. *IEEE Transactions on Signal Processing*, 46(6):1665–1675.

Hu, G., Yan, F., Kittler, J., Christmas, W., Chan, C. H., Feng, Z., and Huber, P. (2017). Efficient 3D morphable face model fitting. *Pattern Recognition*, 67:366–379.

Hua, C., Wang, H., Lu, S., Liu, C., and Khalid, S. M. (2019). A Novel Method of Building Functional Brain Network Using Deep Learning Algorithm with Application in Proficiency Detection. *International Journal of Neural Systems*, 29(1):1850015.

Huber, P. J. (1998). Chapter 12 Robust statistics. In *Data Handling in Science and Technology*, volume 20, pages 339–377. Wiley, New York.

Hunyadi, L. and Vajk, I. (2014). Constrained quadratic errors-in-variables fitting. *The Visual Computer*, 30(12):1347–1358.

Islam, M. T., Tasciotti, E., and Righetti, R. (2020). Estimation of Vascular Permeability in Irregularly Shaped Cancers Using Ultrasound Poroelastography. *IEEE Transactions on Biomedical Engineering*, 67(4):1083–1096.

Jafari, M. H., Karimi, N., Nasr-Esfahani, E., Samavi, S., Soroushmehr, S. M., Ward, K., and Najarian, K. (2016). Skin lesion segmentation in clinical images using deep learning. In *Proceedings - International Conference on Pattern Recognition*, volume 0, pages 337–342.

Jafari, M. H., Nasr-Esfahani, E., Karimi, N., Soroushmehr, S. M., Samavi, S., and Najarian, K. (2017). Extraction of skin lesions from non-dermoscopic images for surgical excision of melanoma. *International Journal of Computer Assisted Radiology and Surgery*, 12(6):1021–1030.

Jang, Y. K., Kang, B. J., and Park, K. R. (2008). A study on eyelid localization considering image focus for iris recognition. *Pattern Recognition Letters*, 29(11):1698–1704.

Janunts, E., Kannengießer, M., and Langenbucher, A. (2015). Parametric fitting of corneal height data to a biconic surface. *Zeitschrift fur Medizinische Physik*, 25(1):25–35.

Jeeva, S. and Sivabalakrishnan, M. (2015). Survey on background modeling and foreground detection for real time video surveillance. *Procedia Computer Science*, 50:566–571.

Jia, Y., He, Z., Gholipour, A., and Warfield, S. K. (2016). Single Anisotropic 3-D MR Image Upsampling via Overcomplete Dictionary Trained from In-Plane High Resolution Slices. *IEEE Journal of Biomedical and Health Informatics*, 20(6):1552–1561.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, pages 675–678.

Kadavi, P. R., Lee, C. W., and Lee, S. (2018). Application of ensemble-based machine learning models to landslide susceptibility mapping. *Remote Sensing*, 10(8).

Kampanis, N. A., Delis, A. I., Antonopoulou, D. C., and Kozyrakis, G. (2013). A finite element discretization of the standard parabolic equation in generalized boundary fitting coordinates. *Applied Numerical Mathematics*, 67:152–166.

Kanatani, K. (1994). Statistical Bias of Conic Fitting and Renormalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):320–326.

Kanatani, K. (2008). Statistical optimization for geometric fitting: Theoretical accuracy bound and high order error analysis. *International Journal of Computer Vision*, 80(2):167–188.

Kanatani, K. and Rangarajan, P. (2011). Hyper least squares fitting of circles and ellipses. *Computational Statistics and Data Analysis*, 55(6):2197–2208.

Kanatani, K. and Sugaya, Y. (2010). Unified computation of strict maximum likelihood for geometric fitting. *Journal of Mathematical Imaging and Vision*, 38(1):1–13.

Karnesky, R. A., Sudbrack, C. K., and Seidman, D. N. (2007). Best-fit ellipsoids of atom-probe tomographic data to study coalescence of γ' (L12) precipitates in Ni-Al-Cr. *Scripta Materialia*, 57(4):353–356.

Kasimbeyli, R., Ozturk, Z. K., Kasimbeyli, N., Yalcin, G. D., and Erdem, B. I. (2019). Comparison of Some Scalarization Methods in Multiobjective Optimization: Comparison of Scalarization Methods. *Bulletin of the Malaysian Mathematical Sciences Society*, 42(5):1875–1905.

Kerahroodi, M. A., Aubry, A., De Maio, A., Naghsh, M. M., and Modarres-Hashemi, M. (2017). A coordinate-descent framework to design low PSL/ISL sequences. *IEEE Transactions on Signal Processing*, 65(22):5942–5956.

Keren, D., Cooper, D., and Subrahmonia, J. (1994). Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):38–53.

Keren, D. and Gotsman, C. (1999). Fitting curves and surfaces with constrained implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):31–41.

Kim, J., Lee, J. K., and Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 1646–1654.

Kim, S. W., Yun, K., Yi, K. M., Kim, S. J., and Choi, J. Y. (2013). Detection of moving objects with a moving camera using non-panoramic background model. *Machine Vision and Applications*, 24(5):1015–1028.

Konda, K. R., Conci, N., and De Natale, F. (2016). Global Coverage Maximization in PTZ-Camera Networks Based on Visual Quality Assessment. *IEEE Sensors Journal*, 16(16):6317–6332.

Köning, R., Wimmer, G., and Witkovský, V. (2014). Ellipse fitting by nonlinear constraints to demodulate quadrature homodyne interferometer signals and to determine the statistical uncertainty of the interferometric phase. *Measurement Science and Technology*, 25(11):0–11.

Koziarski, M. and Cyganek, B. (2017). Image recognition with deep neural networks in presence of noise - Dealing with and taking advantage of distortions. *Integrated Computer-Aided Engineering*, 24(4):337–349.

Krach, B. and Robertson, P. (2008). Integration of foot-mounted inertial sensors into a bayesian location estimation framework. In *5th Workshop on Positioning, Navigation and Communication 2008, WPNC'08*, pages 55–61.

Kriegman, D. J. and Ponce, J. (1994). Parameterized Families of Polynomials for Bounded Algebraic Curve and Surface Fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):287–303.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Krüger, L. and Wöhler, C. (2011). Accurate chequerboard corner localisation for camera calibration. *Pattern Recognition Letters*, 32(10):1428–1435.

Kubota, T. and Okada, K. (2005). Estimating diameters of pulmonary nodules with competition-diffusion and robust ellipsoid fit. In Liu, Y., Jiang, T., and Zhang, C., editors, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3765 LNCS of *Lecture Notes in Computer Science*, pages 324–334. Springer Berlin Heidelberg.

Kurt, O. and Arslan, O. (2019). A general accuracy measure for quality of elliptic sections fitting. *Measurement: Journal of the International Measurement Confederation*, 145:640–647.

Kuruoğlu, E. E., Rayner, P. J., and Fitzgerald, W. J. (1997). Least Lp-norm estimation of autoregressive model coefficients of symmetric α-stable processes. *IEEE Signal Processing Letters*, 4(7):201–203.

Ladrón De Guevara, I., Muñoz, J., De Cózar, O. D., and Blázquez, E. B. (2011). Robust fitting of circle arcs. *Journal of Mathematical Imaging and Vision*, 40(2):147–161.

Landman, B. A., Huang, A. J., Gifford, A., Vikram, D. S., Lim, I. A. L., Farrell, J. A. D., Bogovic, J. A., Hua, J., Chen, M., Jarso, S., and Others (2011). Multi-parametric neuroimaging reproducibility: a 3-T resource study. *Neuroimage*, 54(4):2854–2866.

Lauterbur, P. C. (1973). Image formation by induced local interactions: Examples employing nuclear magnetic resonance. *Nature*, 242(5394):190–191.

Lauterbur, P. C., Mansfield, P., and Plein, S. (2019). Paul Lauterbur and Sir Peter Mansfield for MRI. *European Heart Journal*, 40(24):1898–1899.

Lee, D. and McDaniel, S. T. (1987). Ocean Acoustic Propagation By Finite Difference Methods. *Computers & mathematics with applications*, 14(5).

Lee, D. T. and Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242.

Lee, P. Y. and Moore, J. B. (2004). Geometric optimization for 3D pose estimation of quadratic surfaces. In *Conference Record - Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 131–135.

Leedan, Y. and Meer, P. (2000). Heteroscedastic regression in computer vision: problems with bilinear constraint. *International Journal of Computer Vision*, 37(2):127–150.

Lehmann, T. M., Gönner, C., and Spitzer, K. (1999). Survey: Interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075.

Li, C., Dai, B., Wang, R., Fang, Y., Yuan, X., and Wu, T. (2016). Multilane detection based on omnidirectional camera using anisotropic steerable filters. *IET Intelligent Transport Systems*, 10(5):298–307.

Li, H. (2019). Multiple ellipse fitting of densely connected contours. *Information Sciences*, 502:330–345.

Li, H., He, F., Liang, Y., and Quan, Q. (2020). A dividing-based many-objective evolutionary algorithm for large-scale feature selection. *Soft Computing*, 24(9):6851–6870.

Li, P., Zhang, C., and Jia, S. (2012). A new method of fitting implicit conic to plane scattered data points. *Computer-Aided Design and Applications*, 9(2):147–154.

Li, Q. and Griffiths, J. G. (2004). Least squares ellipsoid specific fitting. In *Proceedings - Geometric Modeling and Processing 2004*, pages 335–340.

Li, Q., Li, Z., Zhang, Y., Yin, G., and Li, J. (2018a). Integrated calibration of magnetic gradient tensor system based on ellipsoid fitting. *Zhongguo Guanxing Jishu Xuebao/Journal of Chinese Inertial Technology*, 26(2):187–195.

Li, Y., Hu, J., Zhao, X., Xie, W., and Li, J. J. (2017). Hyperspectral image super-resolution using deep convolutional neural network. *Neurocomputing*, 266:29–41.

Li, Z., Tang, J., and He, X. (2018b). Robust structured nonnegative matrix factorization for image representation. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1947–1960.

Li, Z., Yu, J., Wang, Y., Zhou, H., Yang, H., and Qiao, Z. (2019). DeepVolume: Brain Structure and Spatial Connection-Aware Network for Brain MRI Super-Resolution. *IEEE Transactions on Cybernetics*, pages 1–14.

Liang, J., Li, P., Zhou, D., So, H. C., Liu, D., Leung, C. S., and Sui, L. (2019). Robust ellipse fitting via alternating direction method of multipliers. *Signal Processing*, 164:30–40.

Liang, J., Zhang, M., Liu, D., Zeng, X., Ojowu, O., Zhao, K., Li, Z., and Liu, H. (2013). Robust ellipse fitting based on sparse combination of data points. *IEEE Transactions on Image Processing*, 22(6):2207–2218.

Liang, Y., Wang, J., Zhou, S., Gong, Y., and Zheng, N. (2016). Incorporating image priors with deep convolutional neural networks for image super-resolution. *Neurocomputing*, 194:340–347.

Liao, M., Zhao, Y. Q., Zeng, Y. Z., Huang, Z. C., Zhang, B. K., and Zou, B. J. (2017). Automatic segmentation for cell images based on support vector machine and ellipse fitting. *Zhejiang Daxue Xuebao (Gongxue Ban)/Journal of Zhejiang University (Engineering Science)*, 51(4):722–728.

Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (2017). Enhanced Deep Residual Networks for Single Image Super-Resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:1132–1140.

Lin, G., Wu, Q., Qiu, L., and Huang, X. (2018). Image super-resolution using a dilated convolutional neural network. *Neurocomputing*, 275:1219–1230.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88.

Liu, H., Yang, C., Pan, N., Song, E., and Green, R. (2010). Denoising 3D MR images by the enhanced non-local means filter for Rician noise. *Magnetic Resonance Imaging*, 28(10):1485–1496.

Liu, H., Zhang, C., and Liang, X. (2004). Fitting conic to scattered data in a plane. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal of Computer-Aided Design and Computer Graphics*, 16(11):1594–1596.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234(November 2016):11–26.

Lopes, M. E. (2019). Estimating the algorithmic variance of randomized ensembles via the bootstrap. *Annals of Statistics*, 47(2):1088–1112.

López-Rubio, E. (2016). Superresolution from a single noisy image by the median filter transform. *SIAM Journal on Imaging Sciences*, 9(1):82–115.

López-Rubio, E., Thurnhofer-Hemsi, K., Blázquez-Parra, E. B., de Cózar-Macías, Ó. D., and Ladrón-de Guevara-Muñoz, M. C. (2018). A fast robust geometric fitting method for parabolic curves. *Pattern Recognition*, 84:301–316.

López-Rubio, E., Thurnhofer-Hemsi, K., de Cózar-Macías, Ó. D., Blázquez-Parra, E. B., Muñoz-Pérez, J., and Ladrón-de-Guevara-López, I. (2017). Robust Fitting of Ellipsoids by Separating Interior and Exterior Points During Optimization. *Journal of Mathematical Imaging and Vision*, 58(2):189–210.

López-Rubio, F. J. and López-Rubio, E. (2015). Features for stochastic approximation based foreground detection. *Computer Vision and Image Understanding*, 133:30–50.

Lötjönen, J. M., Wolz, R., Koikkalainen, J. R., Thurfjell, L., Waldemar, G., Soininen, H., and Rueckert, D. (2010). Fast and robust multi-atlas segmentation of brain magnetic resonance images. *NeuroImage*, 49(3):2352–2365.

Lou, X., Zhou, L., and Jia, Y. (2015). Realization of ellipsoid fitting calibration for three-axis magnetic sensor based on STM32 embedded system. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8944 of *Lecture Notes in Computer Science*, pages 717–726. Springer International Publishing.

Lucas, B. D. and Kanade, T. (1981). Iterative Image Registration Technique With an Application To Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, volume 2, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lukács, G., Martin, R., and Marshall, D. (1998). Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In Burkhardt, H. and Neumann, B., editors, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1406 of *Lecture Notes in Computer Science*, pages 671–686. Springer Berlin Heidelberg.

Luque-Baena, R. M., López-Rubio, E., Domínguez, E., Palomo, E. J., and Jerez, J. M. (2015). A self-organizing map to improve vehicle detection in flow monitoring systems. *Soft Computing*, 19(9):2499–2509.

Ma, Y., Genton, M. G., and Parzen, E. (2011). Asymptotic properties of sample quantiles of discrete distributions. *Annals of the Institute of Statistical Mathematics*, 63(2):227–243.

Macovski, A. (1996). Noise in MRI. *Magnetic Resonance in Medicine*, 36(3):494–497.

Maggioni, M., Katkovnik, V., Egiazarian, K., and Foi, A. (2013). Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133.

Magoulès, F. (2011). *Computational fluid dynamics*. McGraw-Hill, New York.

Mahdavi, S. and Salcudean, S. E. (2008). 3D prostate segmentation based on ellipsoid fitting, image tapering and warping. In *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - "Personalized Healthcare through Technology"*, pages 2988–2991.

Malyugina, A., Igudesman, K., and Chickrin, D. (2014). Least-squares fitting of a three-dimensional ellipsoid to noisy data. *Applied Mathematical Sciences*, 8(149-152):7409–7421.

Man, Y. and Li, H. (2014). Focal length precise measurement method for optics system based on lunar imaging. In *International Symposium on Optoelectronic Technology and Application 2014: Image Processing and Pattern Recognition*, volume 9301, page 93010M.

Manjón, J. V., Coupé, P., Buades, A., Collins, D. L., and Robles, M. (2010a). MRI Superresolution Using Self-Similarity and Image Priors. *International Journal of Biomedical Imaging*, 2010:1–11.

Manjón, J. V., Coupé, P., Buades, A., Fonov, V., Collins, D. L., and Robles, M. (2010b). Non-local MRI upsampling. *Medical Image Analysis*, 14(6):784–792.

Manjón, J. V., Coupé, P., Buades, A., Louis Collins, D., and Robles, M. (2012). New methods for MRI denoising based on sparseness and self-similarity. *Medical Image Analysis*, 16(1):18–27.

Manjón, J. V., Coupé, P., Martí-Bonmatí, L., Collins, D. L., and Robles, M. (2010c). Adaptive non-local means denoising of MR images with spatially varying noise levels. *Journal of Magnetic Resonance Imaging*, 31(1):192–203.

Marcus, D. S., Harwell, J., Olsen, T., Hodge, M., Glasser, M. F., Prior, F., Jenkinson, M., Laumann, T., Curtiss, S. W., and Van Essen, D. C. (2011). Informatics and data mining tools and strategies for the human connectome project. *Frontiers in Neuroinformatics*, 5:4.

Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C., and Buckner, R. L. (2007). Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *Journal of Cognitive Neuroscience*, 19(9):1498–1507.

Maronna, R. A. and Martin, R. D. (2019). *Robust Statistics: Theory and Methods (with R)*. John Wiley & Sons, 2nd edition.

Martinetz, T. and Schulten, K. (1991). A "neural-gas" network learns topologies. Artificial Neural Networks 1, 397–402 (1991). In Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Proceedings of the 1991 International Conference, ICANN-91*, volume 1, pages 397–402. Elsevier.

Martinetz, T. M., Berkovich, S. G., and Schulten, K. J. (1993). "Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569.

Matei, B. C. and Meer, P. (2006). Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1537–1552.

Mathews, V. J. and Xie, Z. (1993). A Stochastic Gradient Adaptive Filter with Gradient Adaptive Step Size. *IEEE Transactions on Signal Processing*, 41(6):2075–2087.

McCann, M. T., Jin, K. H., and Unser, M. (2017). Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95.

McConnell Brain Imaging Centre (2006). BrainWeb: Simulated Brain Database. http://mouldy.bic.mni.mcgill.ca/brainweb/.

Meer, P., Mintz, D., Rosenfeld, A., and Kim, D. Y. (1991). Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70.

Mirajkar, G., Patil, S., and Pawar, M. (2012). Skull Stripping Using Geodesic Active Contours in Magnetic Resonance Images. In *2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks*, pages 301–306. IEEE.

Mirzaei, G. and Adeli, H. (2019). Segmentation and clustering in brain MRI imaging. *Reviews in the Neurosciences*, 30(1):31–44.

Misztal, K. and Tabor, J. (2016). Ellipticity and Circularity Measuring via Kullback–Leibler Divergence. *Journal of Mathematical Imaging and Vision*, 55(1):136–150.

Mitchell, D. R. and Van den Berg, J. A. (2016). Development of an ellipse fitting method with which to analyse selected area electron diffraction patterns. *Ultramicroscopy*, 160:140–145.

Molina-Cabello, M. A., Luque-Baena, R. M., López-Rubio, E., and Thurnhofer-Hemsi, K. (2018). Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images. *Integrated Computer-Aided Engineering*, 25(4):321–333.

Muhamad, A.-h., Nezar, W., and Azzaldeen, A. (2015). The curve of dental arch in normal occlusion The curve of dental arch in normal occlusion. *Open Science Journal of Clinical Medicine*, 3(2):47–54.

Muñoz-Pérez, J., De Cózar-Macías, O. D., Blázquez-Parra, E. B., and Ladrón De Guevara-López, I. (2014). Multicriteria robust fitting of elliptical primitives. *Journal of Mathematical Imaging and Vision*, 49(2):492–509.

Narote, S. P., Bhujbal, P. N., Narote, A. S., and Dhane, D. M. (2018). A review of recent advances in lane detection and departure warning system. *Pattern Recognition*, 73:216–234.

Nida, N., Irtaza, A., Javed, A., Yousaf, M. H., and Mahmood, M. T. (2019). Melanoma lesion detection and segmentation using deep region based convolutional neural network and fuzzy C-means clustering. *International Journal of Medical Informatics*, 124:37–48.

Niu, J., Lu, J., Xu, M., Lv, P., and Zhao, X. (2016). Robust Lane Detection using Two-stage Feature Extraction with Curve Fitting. *Pattern Recognition*, 59:225–233.

Nurunnabi, A., Sadahiro, Y., and Laefer, D. F. (2018). Robust statistical approaches for circle fitting in laser scanning three-dimensional point cloud data. *Pattern Recognition*, 81:417–431.

Ogul, B. B., Sümer, E., and Ogul, H. (2015). Unsupervised Rib Delineation in Chest Radiographs by an Integrative Approach. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, volume 1, pages 260–265, Berlin, Germany. VISAPP, SCITEPRESS - Science and and Technology Publications.

Olivares, A., Ruiz-Garcia, G., Olivares, G., Górriz, J. M., and Ramirez, J. (2013). Automatic determination of validity of input data used in ellipsoid fitting MARG calibration algorithms. *Sensors (Switzerland)*, 13(9):11797–11817.

Oliveira, J. P., Bioucas-Dias, J. M., and Figueiredo, M. A. (2009). Adaptive total variation image deblurring: A majorization-minimization approach. *Signal Processing*, 89(9):1683–1693.

Oliveira, R. B., Papa, J. P., Pereira, A. S., and Tavares, J. M. R. S. (2018). Computational methods for pigmented skin lesion classification in images: review and future trends. *Neural Computing and Applications*, 29(3):613–636.

Ortega-Zamorano, F., Jerez, J. M., Gómez, I., and Franco, L. (2017). Layer multiplexing FPGA implementation for deep back-propagation learning. *Integrated Computer-Aided Engineering*, 24(2):171–185.

Ozan, E. C., Kiranyaz, S., and Gabbouj, M. (2016). Joint K-Means quantization for Approximate Nearest Neighbor Search. *Proceedings - International Conference on Pattern Recognition*, 0(11):3645–3649.

Pal, A., Garain, U., Chandra, A., Chatterjee, R., and Senapati, S. (2018). Psoriasis skin biopsy image segmentation using Deep Convolutional Neural Network. *Computer Methods and Programs in Biomedicine*, 159:59–69.

Pal, N. R. and Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294.

Pan, M. S., Yang, X. L., and Tang, J. T. (2012). Research on interpolation methods in medical image processing. *Journal of Medical Systems*, 36(2):777–807.

Park, J., Hwang, D., Kim, K. Y., Kang, S. K., Kim, Y. K., and Lee, J. S. (2018). Computed tomography super-resolution using deep convolutional neural network. *Physics in Medicine and Biology*, 63(14).

Park, S. and Kwak, N. (2018). Independent component analysis by lp-norm optimization. *Pattern Recognition*, 76:752–760.

Park, S. K. and Suh, Y. S. (2010). A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems. *Sensors (Switzerland)*, 10(10):9163–9178.

Paschal, C. B., Morris, H. D., Paschal, C. B., and Morris, H. D. (2017). K-Space in the Clinic K-Space in the Clinic. *Journal of Magnetic Resonance Imaging*, 19(February 2004):145–159.

Patel, V., Dinov, I. D., Van Horn, J. D., Thompson, P. M., and Toga, A. W. (2010). LONI MiND: Metadata in NIfTI for DWI. *NeuroImage*, 51(2):665–676.

Patil, U., Tabib, R. A., Konin, C. M., and Mudenagudi, U. (2018). Evidence-based framework for multi-image super-resolution. *Advances in Intelligent Systems and Computing*, 709:413–423.

Paton, K. (1970). Conic sections in chromosome analysis. *Pattern Recognition*, 2(1):39–40.

Peng Chen, H., Shen, X. J., and Long, J. W. (2016). Histogram-based colour image fuzzy clustering algorithm. *Multimedia Tools and Applications*, 75(18):11417–11432.

Pereira Dos Santos, F. and Antonelli Ponti, M. (2019). Robust Feature Spaces from Pre-Trained Deep Network Layers for Skin Lesion Classification. In *Proceedings - 31st Conference on Graphics, Patterns and Images, SIBGRAPI 2018*, pages 189–196. IEEE.

Pham, C. H., Ducournau, A., Fablet, R., and Rousseau, F. (2017). Brain MRI super-resolution using deep 3D convolutional networks. In *Proceedings - International Symposium on Biomedical Imaging*, pages 197–200.

Pham, C.-H., Tor-Díez, C., Meunier, H., Bednarek, N., Fablet, R., Passat, N., and Rousseau, F. (2019). Multiscale brain MRI super-resolution using deep 3D convolutional networks. *Computerized Medical Imaging and Graphics*, 77:101647.

Pham, H. and Olafsson, S. (2019). Bagged ensembles with tunable parameters. *Computational Intelligence*, 35(1):184–203.

Pitts, W. and McCulloch, W. S. (1947). How we know universals the perception of auditory and visual forms. *The Bulletin of Mathematical Biophysics*, 9(3):127–147.

Prasad, D. K., Leung, M. K., and Quek, C. (2013). ElliFit: An unconstrained, non-iterative, least squares based geometric Ellipse Fitting method. *Pattern Recognition*, 46(5):1449–1465.

Pratt, V. (1987). Direct least-squares fitting of algebraic surfaces. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, 21(4):145–152.

Prince, J. L., Carass, A., Zhao, C., Dewey, B. E., Roy, S., and Pham, D. L. (2019). Image synthesis and superresolution in medical imaging. In Zhou, S. K., Rueckert, D., and Fichtinger, G., editors, *Handbook of Medical Image Computing and Computer Assisted Intervention*, pages 1–24. Academic Press.

Rafiei, M. H. and Adeli, H. (2017). A New Neural Dynamic Classification Algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12):3074–3083.

Raspet, R. and Di, X. (1990). Calculation of turbulence effects in an upward-refracting atmosphere. *Journal of the Acoustical Society of America*, 87(6):2428–2437.

Ray, A. and Srivastava, D. C. (2008). Non-linear least squares ellipse fitting using the genetic algorithm with applications to strain analysis. *Journal of Structural Geology*, 30(12):1593–1602.

Rimon, E. and Boyd, S. P. (1997). Obstacle Collision Detection Using Best Ellipsoid Fit. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 18(2):105–126.

Robb, R. A., Hanson, D. P., Karwoski, R. A., Larson, A. G., Workman, E. L., and Stacy, M. C. (1989). Analyze: A Comprehensive, operator-interactive software package for multidimensional medical image display and analysis. *Computerized Medical Imaging and Graphics*, 13(6):433–454.

Robertson, J. S., Schlatter, P. J., and Siegmann, W. L. (1996). Sound propagation over impedance discontinuities with the parabolic approximation. *The Journal of the Acoustical Society of America*, 99(2):761–767.

Robertson, J. S., Siegmann, W. L., and Jacobson, M. J. (1995). Low-frequency sound propagation modeling over a locally reacting boundary with the parabolic approximation. *The Journal of the Acoustical Society of America*, 98(2):1130–1137.

Rodríguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A New classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9351 of *LNCS*, pages 234–241. Springer.

Rosin, P. L. (1993a). A note on the least squares fitting of ellipses. *Pattern Recognition Letters*, 14(10):799–808.

Rosin, P. L. (1993b). Ellipse fitting by accumulating five-point fits. *Pattern Recognition Letters*, 14(8):661–669.

Rosin, P. L. (1996). Analysing error of fit functions for ellipses. *Pattern Recognition Letters*, 17(14):1461–1470.

Rosin, P. L. (1999). Further five-point fit ellipse fitting. *Graphical Models and Image Processing*, 61(5):245–259.

Roth, G. and Levine, M. D. (1993). Extracting geometric primitives. *Computer Vision and Image Understanding*, 58(1):1–22.

Rother, C., Kolmogorov, V., and Blake, A. (2004). GrabCut - Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers, SIGGRAPH 2004*, volume 23, pages 309–314. ACM.

Rousseau, F. (2008). Brain Hallucination. *Computer Vision–ECCV 2008*, pages 497–508.

Rueda, A., Malpica, N., and Romero, E. (2013). Single-image super-resolution of brain MR images using overcomplete dictionaries. *Medical Image Analysis*, 17(1):113–132.

Sahin, T. and Unel, M. (2008). Stable algebraic surfaces for 3D object representation. *Journal of Mathematical Imaging and Vision*, 32(2):127–137.

Salomons, E. M. (1998). Improved Green's function parabolic equation method for atmospheric sound propagation. *The Journal of the Acoustical Society of America*, 104(1):100–111.

Samarasinghe, G., Sowmya, A., and Moses, D. A. (2015). A semi-quantitative analysis model with parabolic modelling for DCE-MRI sequences of prostate. In *2014 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2014*. IEEE.

Samet, H. (1984). The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260.

Samper, A., González, G., and Herrera, B. (2017). Determination of the geometric shape which best fits an architectural arch within each of the conical curve types and hyperbolic-cosine curve types: The case of Palau Güell by Antoni Gaudí. *Journal of Cultural Heritage*, 25:56–64.

Sampson, P. D. (1982). Fitting conic sections to "very scattered" data: An iterative refinement of the bookstein algorithm. *Computer Graphics and Image Processing*, 18(1):97–108.

Sappa, A. D. and Rouhani, M. (2009). Efficient distance estimation for fitting implicit quadric surfaces. In *Proceedings - International Conference on Image Processing, ICIP*, pages 3521–3524.

Sarakon, P., Charoenpong, T., and Charoensiriwath, S. (2013). Nose tip detection using ellipsoid fitting for 2.5D partial face data. In *BMEiCON 2013 - 6th Biomedical Engineering International Conference*, pages 1–5.

Sarakon, P., Charoenpong, T., and Charoensiriwath, S. (2014). Face shape classification from 3D human data by using SVM. In *BMEiCON 2014 - 7th Biomedical Engineering International Conference*, pages 1–5.

Schapire, R. E. (2003). The Boosting Approach to Machine Learning: An Overview. In *Nonlinear estimation and classification*, pages 149–171. Springer.

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Seheult, A. H., Green, P. J., Rousseeuw, P. J., and Leroy, A. M. (1989). Robust Regression and Outlier Detection. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 152(1):133.

Shah, A. J. and Gupta, S. B. (2012). Image super resolution - A survey. In *Proceedings on 2012 1st International Conference on Emerging Technology Trends in Electronics, Communication and Networking, ET2ECN 2012*, pages 1–6.

Shahin, A. H., Kamal, A., and Elattar, M. A. (2019). Deep Ensemble Learning for Skin Lesion Classification from Dermoscopic Images. In *2018 9th Cairo International Biomedical Engineering Conference, CIBEC 2018 - Proceedings*, pages 150–153. IEEE.

Shi, F., Cheng, J., Wang, L., Yap, P. T., and Shen, D. (2015). LRTV: MR Image Super-Resolution with Low-Rank and Total Variation Regularizations. *IEEE Transactions on Medical Imaging*, 34(12):2459–2466.

Shi, J., Li, Z., Ying, S., Wang, C., Liu, Q., Zhang, Q., and Yan, P. (2019). MR Image Super-Resolution via Wide Residual Networks with Fixed Skip Connection. *IEEE Journal of Biomedical and Health Informatics*, 23(3):1129–1140.

Shi, J., Liu, Q., Wang, C., Zhang, Q., Ying, S., and Xu, H. (2018). Super-resolution reconstruction of MR image with a novel residual learning network algorithm. *Physics in Medicine and Biology*, 63(8).

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, abs/1409.1556.

Singh, A., Porikli, F., and Ahuja, N. (2014). Super-resolving noisy images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2846–2853.

Skog, I., Nilsson, J. O., and Händel, P. (2010). Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, pages 1–6.

Smith, T. B. and Nayak, K. S. (2010). MRI artifacts and correction strategies. *Imaging in Medicine*, 2(4):445–457.

Sobhani, E., Sadeghi, M., Babaie-Zadeh, M., and Jutten, C. (2017). A robust ellipse fitting algorithm based on Sparsity of outliers. In *25th European Signal Processing Conference, EUSIPCO 2017*, volume 2017-Janua, pages 1195–1199.

Song, L., Mi, H., Lü, Y., and Liu, Q. (2011). Bagging-based System Combination for Domain Adaption. In *MT-Summit-2011*, pages 293–299.

Späth, H. (2001). Least squares fitting with rotated paraboloids. *Mathematical Communications*, 6(2):173–179.

Steuer, R. E. and Choo, E. U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344.

Sturm, P. and Gargallo, P. (2007). Conic fitting using the geometric distance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4844 LNCS(PART 2):784–795.

Su, L., Yang, K., Hu, H., and Yang, Z. (2019). Long-term hydropower generation scheduling of large-scale cascade reservoirs using chaotic adaptive multi-objective bat algorithm. *Water (Switzerland)*, 11(11):2373.

Sullivan, S., Sandford, L., and Ponce, J. (1994). Using Geometric Distance Fits for 3-D Object Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1183–1196.

Sung Joon Ahn, Rauh, W., Hyung Suck Cho, and Warnecke, H.-J. (2002). Orthogonal distance fitting of implicit curves and surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):620–638.

Susi, M., Renaudin, V., and Lachapelle, G. (2013). Motion mode recognition and step detection algorithms for mobile phone users. *Sensors (Switzerland)*, 13(2):1539–1562.

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 07-12-June, pages 1–9. IEEE.

Szpak, Z. L., Chojnacki, W., and van den Hengel, A. (2015). Guaranteed Ellipse Fitting with a Confidence Region and an Uncertainty Measure for Centre, Axes, and Orientation. *Journal of Mathematical Imaging and Vision*, 52(2):173–199.

Taubin, G. (1991). Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge

and Range Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138.

Teti, J. (2001). Parabolic equation methods for electromagnetic wave propagation [Book Review]. *IEEE Antennas and Propagation Magazine*, 43(3):96–97.

Theekapun, C., Hase, H., and Tokai, S. (2006). Robust nose localization by using fitting of ellipse from 2.5d image. In *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, pages 1–4.

Thevenaz, P., Blu, T., and Unser, M. (2000). Interpolation revisited [medical images application]. *IEEE Transactions on Medical Imaging*, 19(7):739–758.

Thurnhofer-Hemsi, K., López-Rubio, E., Blázquez-Parra, E. B., Ladrón-de Guevara-Muñoz, M. C., and De-Cózar-Macias, Ó. D. (2020a). Ellipse fitting by spatial averaging of random ensembles. *Pattern Recognition*, 106:107406.

Thurnhofer-Hemsi, K., Lopez-Rubio, E., Dominguez, E., Luque-Baena, R. M., and Molina-Cabello, M. A. (2017). Panoramic background modeling for PTZ cameras with competitive learning neural networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2017-May, pages 396–403. IEEE.

Thurnhofer-Hemsi, K., López-Rubio, E., Domínguez, E., Luque-Baena, R. M., and Molina-Cabello, M. A. (2018a). Panorama construction for PTZ camera surveillance with the neural gas network. *Expert Systems*, 35(2):e12249.

Thurnhofer-Hemsi, K., López-Rubio, E., Domínguez, E., Luque-Baena, R. M., and Roé-Vellvé, N. (2020b). Deep learning-based super-resolution of 3D magnetic resonance images by regularly spaced shifting. *Neurocomputing*, 398:314–327.

Thurnhofer-Hemsi, K., López-Rubio, E., Roé-Vellvé, N., and Deka, L. (2020c). Super-resolution of 3D MRIs corrupted by heavy noise with the Median Filter Transform. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1–5.

Thurnhofer-Hemsi, K., Lopez-Rubio, E., Roe-Vellve, N., Dominguez, E., and Molina-Cabello, M. A. (2018b). Super-resolution of 3D Magnetic Resonance Images by Random Shifting and Convolutional Neural Networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2018-July.

Thurnhofer-Hemsi, K., López-Rubio, E., Roé-Vellvé, N., and Molina-Cabello, M. A. (2019). Deep Learning Networks with p-norm Loss Layers for Spatial Resolution Enhancement of 3D Medical Images. In Ferrández Vicente, J. M., Álvarez-Sánchez, J. R., de la Paz López, F., Toledo Moreo, J., and Adeli, H., editors, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11487 LNCS, pages 287–296, Cham. Springer International Publishing.

Thurnhofer-Hemsi, K., López-Rubio, E., Roé-Vellvé, N., and Molina-Cabello, M. A. (2020d). Multiobjective optimization of deep neural networks with combinations of Lp-norm cost functions for 3D medical image super-resolution. *Integrated Computer-Aided Engineering*, 27(3):233–251.

Tian, J. and Ma, K. K. (2011). A survey on super-resolution imaging. *Signal, Image and Video Processing*, 5(3):329–342.

Tien Bui, D., Ho, T. C., Pradhan, B., Pham, B. T., Nhu, V. H., and Revhaug, I. (2016). GIS-based modeling of rainfall-induced landslides using data mining-based functional trees classifier with AdaBoost, Bagging, and MultiBoost ensemble frameworks. *Environmental Earth Sciences*, 75(14).

Todd-Pokropek, A., Cradduck, T. D., and Deconinck, F. (1992). A file format for the exchange of nuclear medicine image data: A specification of interfile version 3.3. *Nuclear Medicine Communications*, 13(9):673–699.

Torrente, M. L., Biasotti, S., and Falcidieno, B. (2018). Recognition of feature curves on 3D shapes using an algebraic approach to Hough transforms. *Pattern Recognition*, 73:111–130.

Trinh, D. H., Luong, M., Dibos, F., Rocchisani, J. M., Pham, C. D., and Nguyen, T. Q. (2014). Novel example-based method for super-resolution and denoising of medical images. *IEEE Transactions on Image Processing*, 23(4):1882–1895.

Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, Upper Saddle River/New York.

Tu, C. T., Lin, H. W., Lin, H. J., and Li, Y. S. (2016). Super-Resolution Based on Clustered Examples. *International Journal of Pattern Recognition and Artificial Intelligence*, 30(6).

Uchiyama, T. and Arbib, M. A. (1994). Color Image Segmentation Using Competitive Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1197–1206.

Unser, M., Aldroubi, A., and Eden, M. (1992). On the Asymptotic Convergence of B-spline Wavelets to Gabor Functions. *IEEE Transactions on Information Theory*, 38(2):864–872.

Vajk, I. and Hetthessy, J. (2003). Identification of nonlinear errors-in-variables models cation of nonlinear errors-in-variables models. *Automatica*, 39:2099–2107.

Valente, R. A. and Abrão, T. (2016). MIMO transmit scheme based on morphological perceptron with competitive learning. *Neural Networks*, 80:9–18.

van de Weijer, J., Gevers, T., and Gijsenij, A. (2007). Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214.

Van Den Doel, K. and Ascher, U. (2012). The chaotic nature of faster gradient descent methods. *Journal of Scientific Computing*, 51(3):560–581.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., and Ugurbil, K. (2013). The WU-Minn Human Connectome Project: An overview. *NeuroImage*, 80:62–79.

van Huffel, S. (1997). *Recent Advances in Total Least Squares Techniques and Errors-in-variables Modeling*. Proceedings in Applied Mathematics. Society for Industrial and Applied Mathematics.

Van Reeth, E., Tham, I. W., Tan, C. H., and Poh, C. L. (2012). Super-resolution in magnetic resonance imaging: A review. *Concepts in Magnetic Resonance Part A: Bridging Education and Research*, 40 A(6):306–325.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38(1):49–95.

Várady, T., Martin, R. R., and Cox, J. (1997). Reverse engineering of geometric models - An introduction. *CAD Computer Aided Design*, 29(4):255–268.

Varcheie, P. D. Z. and Bilodeau, G. A. (2011). Adaptive fuzzy particle filter tracker for a PTZ camera in an IP surveillance system. *IEEE Transactions on Instrumentation and Measurement*, 60(2):354–371.

Veredas, F., Mesa, H., and Morente, L. (2010). Binary tissue classification on wound images with neural networks and bayesian classifiers. *IEEE Transactions on Medical Imaging*, 29(2):410–427.

von Kries, J. (1970). Influence of adaptation on the effects produced by luminous stimuli. *Sources of color vision*, 3:109–119.

Waibel, P., Matthes, J., and Gröll, L. (2015). Constrained Ellipse Fitting with Center on a Line. *Journal of Mathematical Imaging and Vision*, 53(3):364–382.

Wang, C., Yan, X., Smith, M., Kochhar, K., Rubin, M., Warren, S. M., Wrobel, J., and Lee, H. (2015). A unified framework for automatic wound segmentation and analysis with deep convolutional neural networks. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, volume 2015-Novem, pages 2415–2418.

Wang, J., Gu, F., Zhang, C., and Zhang, G. (2010). Lane boundary detection based on parabola model. In *2010 IEEE International Conference on Information and Automation, ICIA 2010*, pages 1729–1734.

Wang, P. and Bai, X. (2018). Regional parallel structure based CNN for thermal infrared face identification. *Integrated Computer-Aided Engineering*, 25(3):247–260.

Wang, W., Wang, C., Duan, B., Leng, G., and Li, X. (2014). Compensation for gravity deformation via subreflector motion of 65 m shaped Cassegrain antenna. *IET Microwaves, Antennas and Propagation*, 8(3):158–164.

Wang, Y., Yang, J., Xiao, C., and An, W. (2018). Fast Convergence Strategy for Multi-Image Superresolution via Adaptive Line Search. *IEEE Access*, 6:9129–9139.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Welsch, R. E. (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827.

Worth, A. J. (2010). MGH CMA Internet Brain Segmentation Repository (IBSR). http://www.cma.mgh.harvard.edu/ibsr/.

Xiang, L., Qiao, Y., Nie, D., An, L., Lin, W., Wang, Q., and Shen, D. (2017). Deep auto-context convolutional neural networks for standard-dose PET image estimation from low-dose PET/MRI. *Neurocomputing*, 267:406–416.

Xiao, G., Wang, H., Lai, T., and Suter, D. (2016). Hypergraph modelling for geometric model fitting. *Pattern Recognition*, 60:748–760.

Xiaole, Z., Zhang, H., Liu, H., Qin, Y., Zhang, T., and Zou, X. (2019). Single MR Image Super-Resolution via Channel Splitting and Serial Fusion Network. *arXiv preprint*.

Xie, H., Luo, X., Wang, C., Liu, S., Xu, X., and Tong, X. (2016). Multispectral remote sensing image segmentation using rival penalized controlled competitive learning and fuzzy entropy. *Soft Computing*, 20(12):4709–4722.

Xue, K., Liu, Y., Ogunmakin, G., Chen, J., and Zhang, J. (2013). Panoramic Gaussian mixture model and large-scale range background substraction method for PTZ camera-based surveillance systems. *Machine Vision and Applications*, 24(3):477–492.

Yadav, V. and Kaushik, V. D. (2018). Detection of melanoma skin disease by extracting high level features for skin lesions. *International Journal of Advanced Intelligence Paradigms*, 11(3-4):397–408.

Yan, G., Li, C., Zhou, G., Zhang, W., and Li, X. (2007). Automatic Extraction of Power Lines From Aerial Images. *IEEE Geoscience and Remote Sensing Letters*, 4(3):387–391.

Yang, C.-Y., Ma, C., and Yang, M.-H. (2014). Single-Image Super-Resolution: A Benchmark. In *Proceedings of European Conference on Computer Vision*.

Yang, J., Wright, J., Huang, T. S., and Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873.

Yang, L., Cheng, H., Su, J., and Li, X. (2016). Pixel-to-Model Distance for Robust Background Reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(5):903–916.

Ye, Y. F., Shao, Y. H., Deng, N. Y., Li, C. N., and Hua, X. Y. (2017). Robust Lp-norm least squares support vector regression with feature selection. *Applied Mathematics and Computation*, 305:32–52.

Ying, X., Yang, L., and Zha, H. (2012). A fast algorithm for multidimensional ellipsoid-specific fitting by minimizing a new defined vector norm of residuals using semidefinite programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1856–1863.

Yu, C. Y., Liu, C. C., and Yu, S. S. (2014a). A fovea localization scheme using vessel origin-based parabolic model. *Algorithms*, 7(3):456–470.

Yu, C. Y., Liu, C. C., and Yu, S. S. (2014b). A novel scheme for the fovea localization on retinal images. In Washington, I. C. S., editor, *Proceedings - 2014 International Symposium on Computer, Consumer and Control, IS3C 2014*, pages 609–612.

Yu, J., Kulkarni, S. R., and Poor, H. V. (2009). Robust fitting of ellipses and spheroids. In *Conference Record - Asilomar Conference on Signals, Systems and Computers*, pages 94–98.

Yu, J., Kulkarni, S. R., and Poor, H. V. (2012). Robust ellipse and spheroid fitting. *Pattern Recognition Letters*, 33(5):492–499.

Yu, K., Dang, X., and Chen, Y. (2015). Robustness of the affine equivariant scatter estimator based on the spatial rank covariance matrix. *Communications in Statistics - Theory and Methods*, 44(5):914–932.

Yue, L., Shen, H., Li, J., Yuan, Q., Zhang, H., and Zhang, L. (2016). Image super-resolution: The techniques, applications, and future. *Signal Processing*, 128:389–408.

Zeng, K., Zheng, H., Cai, C., Yang, Y., Zhang, K., and Chen, Z. (2018). Simultaneous single- and multi-contrast super-resolution for brain MRI images based on a convolutional neural network. *Computers in Biology and Medicine*, 99:133–141.

Zhang, C., Li, D., and Tan, J. (2013). The support vector regression with adaptive norms. *Procedia Computer Science*, 18:1730–1736.

Zhang, D., Han, J., Li, C., Wang, J., and Li, X. (2016). Detection of Co-salient Objects by Looking Deep and Wide. *International Journal of Computer Vision*, 120(2):215–232.

Zhang, D., He, J., Zhao, Y., and Du, M. (2015). MR image super-resolution reconstruction using sparse representation, nonlocal similarity and sparse derivative prior. *Computers in Biology and Medicine*, 58:130–145.

Zhang, Y., Comerford, L., Kougioumtzoglou, I. A., and Beer, M. (2018). Lp-norm minimization for stochastic process power spectrum estimation subject to incomplete data. *Mechanical Systems and Signal Processing*, 101:361–376.

Zhang, Z. (1997). Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Zhang, Z. Q. and Yang, G. Z. (2015). Micromagnetometer calibration for accurate orientation estimation. *IEEE Transactions on Biomedical Engineering*, 62(2):553–560.

Zhao, X., Zhang, Y., Zhang, T., and Zou, X. (2019). Channel Splitting Network for Single MR Image Super-Resolution. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 28(11):5649–5662.

Zheng, H., Qu, X., Bai, Z., Liu, Y., Guo, D., Dong, J., Peng, X., and Chen, Z. (2017). Multi-contrast brain magnetic resonance image super-resolution using the local weight similarity. *BMC Medical Imaging*, 17(1):1–13.

Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., and Zhangd, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49.

Zhou, T., Thung, K., Zhu, X., and Shen, D. (2019). Effective feature learning and fusion of multimodality data using stage-wise deep neural network for dementia diagnosis. *Human Brain Mapping*, 40(3):1001–1016.

Tras un largo camino batallador,
dando y recibiendo con amor,
el Dóctor se convirtió en Doctor.

♥ *Dr. Love* ♥